



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

**UNIVERZITA A JEJÍ AREÁLY – INFORMACE A
NAVIGACE (MOBILNÍ APLIKACE)**

THE UNIVERSITY AND ITS CAMPUSES – INFORMATION AND NAVIGATION (MOBILE
APPLICATION)

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Michal Cupák

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Pavel Heriban, Ph.D.

BRNO 2017

Zadání diplomové práce

Ústav: Ústav automatizace a informatiky
Student: **Bc. Michal Cupák**
Studijní program: Strojní inženýrství
Studijní obor: Aplikovaná informatika a řízení
Vedoucí práce: **Ing. Pavel Heriban, Ph.D.**
Akademický rok: 2016/17

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Univerzita a její areály – informace a navigace (mobilní aplikace)

Stručná charakteristika problematiky úkolu:

V prostředí velké univerzity umístěné v mnoha areálech a budovách je orientace osob poměrně obtížná. Vytvářená mobilní aplikace tuto orientaci usnadní. Aplikace poskytne základní informace o jednotlivých lokalitách a bude do nich uživatele navigovat. Data aplikace budou uložena na mobilním zařízení a budou se automaticky aktualizovat v době aktivního internetového připojení.

Cíle diplomové práce:

- 1) Seznámení se s problematikou tvorby mobilních aplikací se zaměřením na multiplatformní GIS aplikace.
- 2) Získání přehledu o dostupných vývojových prostředích.
- 3) Analýza a návrh mobilní aplikace pro usnadnění orientace v prostředí velké univerzity.
- 4) Výběr vhodného vývojového prostředí a implementace navržené aplikace.

Seznam doporučené literatury:

Xamarin. Building Cross Platform Applications. [online]. [cit. 31.10.2016]. Dostupné z: https://developer.xamarin.com/guides/cross-platform/application_fundamentals/building_cross_platform_applications/.

Apple Developer. Guides and Sample Code. Cross Platform. [online]. [cit. 31.10.2016]. Dostupné z: <https://developer.apple.com/library/prerelease/content/navigation/#section=Topics&topic=Cross%20Platform>.

Adobe PhoneGap. PhoneGap Documentation. [online]. [cit. 31.10.2016]. Dostupné z: <http://docs.phonegap.com/>.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2016/17

V Brně, dne

L. S.

doc. Ing. Radomil Matoušek, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Tato práce pojednává o problematice vývoje multiplatformních mobilních aplikací. Součástí práce je návrh a implementace mobilní aplikace pro usnadnění orientace osob v prostředí velké univerzity. Aplikace je navržena pro mobilní operační systémy Android a iOS a je implementována ve vývojovém prostředí Ionic. Aplikace poskytne základní informace o jednotlivých budovách univerzity, zobrazí budovy na mapě a umožní uživatele do těchto budov navigovat.

ABSTRACT

This thesis deals with development of multiplatform mobile applications. Part of the thesis is the design and implementation of a mobile application to facilitate the orientation of people in a large university environment. The application is designed for Android and iOS mobile operating systems and is implemented in the Ionic development environment. The application provides basic information about individual buildings of the university, displays buildings on the map and allows the user to navigate these buildings.

KLÍČOVÁ SLOVA

Multiplatformní mobilní aplikace, orientace v areálu, navigace, Android, iOS, Ionic, Apache Cordova.

KEYWORDS

Multiplatform mobile applications, orientation in the areal, navigation, Android, iOS, Ionic, Apache Cordova.

BIBLIOGRAFICKÁ CITACE

CUPÁK, Michal. *Univerzita a její areály – informace a navigace (mobilní aplikace)*, Brno, 2017. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky.

PODĚKOVÁNÍ

Děkuji vedoucímu mé diplomové práce, Ing. Pavlovi Heribanovi, Ph.D., za odborné vedení, konzultace, množství přínosných podnětů a testování aplikace.

Děkuji také zaměstnancům Oddělení Facility Managementu Masarykovy Univerzity, zejména pak Mgr. Bc. Davidu Miksteinovi, za konzultace návrhu aplikace.

Také děkuji všem dalším testerům, kteří se zapojili do testování aplikace, jmenovitě je to Jan Konzal, Filip Cupák, Filip Mezuláník a Mgr. Pavel Blažek.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. Pavla Heribana, Ph.D. a s použitím literatury uvedené v seznamu literatury.

V Brně dne 26. 5. 2017

.....

Michal Cupák

OBSAH

1	ÚVOD.....	15
2	MULTIPLATFORMNÍ MOBILNÍ APLIKACE	17
2.1	Vývoj mobilních aplikací	17
2.1.1	Nativní mobilní aplikace	17
2.1.2	Hybridní aplikace	18
2.1.3	Mobilní weby.....	19
2.1.4	Srovnání přístupů k vývoji	19
2.2	Geografické informační systémy v mobilních aplikacích	21
3	PŘEHLED DOSTUPNÝCH VÝVOJOVÝCH PROSTŘEDÍ	23
3.1	Nativní vývojová prostředí	23
3.1.1	Android Studio	23
3.1.2	Xcode.....	24
3.1.3	Visual Studio	25
3.2	Multiplatformní (hybridní) vývojová prostředí	26
3.2.1	Xamarin	26
3.2.2	Apache Cordova	27
3.2.3	PhoneGap.....	31
3.2.4	Ionic	31
3.2.5	React Native	33
3.2.6	Appcelerator Titanium.....	34
3.2.7	Sencha Ext JS	35
3.2.8	AppStudio for ArcGIS	35
4	NÁVRH MOBILNÍ APLIKACE	39
4.1	Konkurenční aplikace	39
4.1.1	Budovy MU	39
4.1.2	MUNI Buildings	40
4.1.3	Easy Brno	42
4.2	Požadavky.....	43
4.3	Analýza a návrh aplikace.....	43
4.3.1	Jazyk aplikace.....	43
4.3.2	Obrazovky	44
4.4	Aktualizace dat aplikace	47
4.4.1	Mobilní aplikace	47
4.4.2	Databáze	47
4.4.3	Webový server.....	48
5	IMPLEMENTACE APLIKACE.....	49
5.1	Výběr vhodného vývojového prostředí	49
5.1.1	Instalace SDK mobilních platform	49
5.1.2	Instalace Ionicu.....	49
5.1.3	Práce s Ionicem.....	50
5.2	Implementace navržené aplikace.....	51
5.2.1	Struktura projektu	51
5.2.2	Mapová API.....	52
5.2.3	Lokální databáze.....	52
5.2.4	Aktualizace dat	52

5.3	Implementace serverových služeb	53
5.4	Popis hlavní databáze	54
5.5	Publikování aplikace	54
5.6	Výsledná aplikace	55
5.7	Testování	56
5.7.1	Android	56
5.7.2	iOS	57
6	ZHODNOCENÍ A DISKUZE	59
7	ZÁVĚR.....	61
8	SEZNAM POUŽITÉ LITERATURY	63
9	SEZNAM ZKRATEK, SYMBOLŮ, OBRÁZKŮ A TABULEK.....	65
9.1	Seznam zkratk	65
9.2	Seznam obrázků	65
9.3	Seznam tabulek	66
10	SEZNAM PŘÍLOH	67

1 ÚVOD

Vyznat se v prostředí velké univerzity umístěné v mnoha areálech a budovách může být často velice obtížné. Cílem této práce je vytvoření mobilní aplikace, která usnadní orientaci v těchto rozsáhlých areálech univerzit. Podnět pro vytvoření této mobilní aplikace vneslo Oddělení facility managementu Masarykovy Univerzity s cílem nahradit dvě starší aplikace vydané na tomto oddělení.

Vývoj mobilních aplikací je ve světě informačních technologií velice mladým odvětvím. První mobilní aplikace se objevily v roce 2007 po představení mobilního operačního systému iOS. Od tohoto roku jde vývoj v oblasti mobilních telefonů stále velmi rychle kupředu. Tento vývoj se netýká pouze hardware mobilních zařízení a jejich operačních systémů, týká se také samotných aplikací a způsobu, jak tyto aplikace vyvíjet. V posledních letech se stále více naráží na problém nemultiplatformnosti aplikací, tedy že jedna aplikace není dostupná pro různé platformy. Řešením tohoto problému je vznik nových způsobů vývoje aplikací jako např. vývoj hybridních aplikací. Tato problematika je podrobněji popsána v kapitole 2.

Spolu se vznikajícími novými způsoby vývoje aplikací vzniká také mnoho nových vývojových prostředí pro multiplatformní vývoj mobilních aplikací. V kapitole 3 je uveden základní přehled vývojových prostředí. Jsou zmíněna typická prostředí pro nativní vývoj i prostředí pro vývoj multiplatformních aplikací.

V kapitole 4 jsou představeny již existující mobilní aplikace, které řeší stejný či podobný úkol – usnadnění orientace v prostředí velké univerzity či města. Na základě těchto mobilních aplikací je poté v této kapitole provedena analýza a návrh výsledné aplikace. Aplikace poskytne základní informace o jednotlivých budovách, zobrazí budovy na mapě a umožní uživatele do těchto budov navigovat. Data aplikace jsou uložena přímo v mobilním zařízení, aplikace je tedy schopná běžně fungovat i bez aktivního internetového připojení. Aktualizace dat je v aplikaci zajištěna pomocí online databáze stále dostupné přes webové služby. V době aktivního internetového připojení aplikace zkontroluje dostupnost aktuálnějších dat a tato data případně nabídne uživateli ke stažení.

Pro implementaci navržené mobilní aplikace autor zvolil vývojové prostředí Ionic, které je založeno na webových technologiích HTML, CSS a JavaScript. Tento výběr proběhl na základě přehledu vývojových prostředí uvedených v kapitole 3 s ohledem na vlastní autorovy zkušenosti s webovými technologiemi. Výběr vývojového prostředí Ionic včetně popisu instalace všech nutných součástí a popisu práce v tomto prostředí je dále popsán v kapitole 5. V této kapitole je také uveden popis samotné implementace navržené mobilní aplikace a serverových služeb. V závěru kapitoly je provedeno testování výsledné mobilní aplikace.

V kapitole 6 je výsledná mobilní aplikace zhodnocena a srovnána s aplikacemi uvedenými v kapitole 4. Dále jsou v této kapitole zmíněny návrhy na rozšíření implementované aplikace a budoucí plány s aplikací.

2 MULTIPLATFORMNÍ MOBILNÍ APLIKACE

V této kapitole je uvedeno základní seznámení s problematikou tvorby mobilních aplikací se zaměřením na multiplatformní GIS aplikace. Každá aplikace, která má být úspěšná, musí splňovat dvě základní pravidla: musí být použitelná a univerzální. Použitelná z hlediska uživatelského a univerzální z hlediska ekonomického [1].

2.1 Vývoj mobilních aplikací

Po představení prvního iPhone s operačním systémem iOS v roce 2007 a následném vzniku AppStoru se ve světě IT otevřelo zcela nové odvětví – vývoj mobilních aplikací. K systému iOS se brzy připojila platforma Android, kterou následovaly vylepšené mobilní Windows a další mobilní systémy. S novým hardwarem se postupně vylepšují nejen aplikace samotné, ale i způsob jejich vývoje. Možných přístupů k vývoji mobilních aplikací existuje v dnešní době spousta. Pokud se pomine množství různých vývojových platforem a zaměří se pouze na způsob, jak aplikace vyvíjet, zůstanou 3 základní přístupy k vývoji:

- nativní mobilní aplikace,
- hybridní aplikace,
- mobilní weby.

Každý z těchto přístupů k vývoji je svým pojetím specifický a slouží k jinému použití a typu aplikací. Výběr správného způsobu vývoje by měl být vždy důkladně promyšlen ještě před započatím samotného vývoje aplikace. Tento výběr však není vždy úplně snadný. Zpočátku může být vybraný způsob vývoje velkým pomocníkem a může se zdát jako nejlepší a jediná možná volba. Po čase však může zapříčinit velké problémy. Těmto problémům se vývojář vyvaruje pouze důkladnou analýzou připravované aplikace a promyšlenou volbou přístupu k vývoji.

Další velice důležitou otázkou před započatím každého vývoje mobilních aplikací musí být, jakému okruhu uživatelů bude aplikace určena. A tedy, pro jaké platformy bude aplikace vydána. V dnešní době jsou hlavní platformy tři, a to Android (Google), iOS (Apple) a Windows Phone (Microsoft).

V následujících podkapitolách budou představeny výše zmíněné přístupy k vývoji mobilních aplikací, včetně shrnutí výhod a nevýhod jejich použití [1] [2] [3].

2.1.1 Nativní mobilní aplikace

Vývoj nativních mobilních aplikací je v dnešní době nejrozšířenějším přístupem k vývoji mobilních aplikací. Tento přístup doporučují i samotní vývojáři platforem. Nativní mobilní aplikace přinášejí vývojáři největší možnosti a také umožňují nejlepší výkon výsledné aplikace.

Nativní aplikace se vyvíjí v programovacích jazycích typických pro danou platformu:

- Java pro Android,
- ObjectiveC/Swift pro Apple,
- C#/C++ pro Microsoft.

Vyvinutá nativní aplikace je určena právě pouze pro danou platformu. Tato „nemultiplatformnost“ je zásadním problémem nativních aplikací. Při vývoji nativní aplikace pro tři hlavní platformy (Android, iOS, Windows Phone) je tedy nutné to stejné implementovat třikrát jinak. To ve většině případů znamená „třikrát zaplatit“ za jednu aplikaci. Při následné úpravě nebo aktualizaci aplikace nastává tentýž problém znovu. Z ekonomického hlediska tedy tento přístup výhodný není.

Výhody nativních mobilních aplikací

- výkon – aplikace jsou napsány přímo v jazyce dané platformy, jsou překládány a optimalizovány přímo pro danou platformu,
- nativní UI prvky – vývojový software přímo podporuje grafické prvky uživatelského rozhraní v telefonu,
- podpora nejnovějších API – ovládání všeho, co telefon nabízí.

Nevýhody nativního vývoje

- nemultiplatformnost:
 - vysoká cena,
 - nároky na programátory,
 - náročné opravy aplikací.

2.1.2 Hybridní aplikace

Základem hybridních aplikací je vždy instance prohlížeče, odborně nazvaná WebView. Toto WebView dnes obsahuje každá platforma. Jedná se vlastně o obrazovku, která za pomoci nainstalovaného prohlížeče zobrazuje pro telefony optimalizovaný webový obsah v mobilní aplikaci.

Hybridní aplikace jsou tedy založené na jazycích HTML, CSS a JavaScript. V poslední době tento přístup k vývoji mobilních aplikací zažívá výrazný rozmach. Technologie používané pro vývoj hybridních aplikací prošly za poslední roky výrazným pokrokem a zaznamenaly značná vylepšení. Nejsilnější stránkou tohoto přístupu oproti nativním aplikacím je rychlost vývoje – většina vývoje probíhá pro všechny platformy souběžně.

Výhody hybridních aplikací

- multiplatformnost – vývoj pro všechny platformy současně; s tím souvisí i náklady na vývoj,
- snadné testování – většinu práce lze simulovat v prohlížeči,

- webové technologie – možnost využití existujících znalostí z vývoje webových stránek,
- rychlá implementace, rychlé rozšíření o nové funkce,
- snadná údržba aplikace pro nové verze systémů.

Nevýhody hybridních aplikací

- vyšší nároky na hardware – ze zvyšujícím se výkonem mobilních zařízení a prohlížečů se tento problém minimalizuje, avšak nároky na výkon budou u hybridních aplikací vždy vyšší než u nativních,
- přístup k funkcím telefonu – možná nedostupnost nových nebo méně používaných API,
- chybí nativní UI prvky – problém lze minimalizovat použitím různých „frameworků“ (aplikačních rámců) pro uživatelská rozhraní,
- omezené či vůbec žádné možnosti při úpravě a zpracování audia, videa, obrázků, složité a 3D grafiky.

2.1.3 Mobilní weby

V tomto přístupu se nejedná o plnohodnotnou mobilní aplikaci, nýbrž jen o dobře optimalizovaný mobilní web. Mobilní weby jsou zajímavé v případech, kdy by mobilní aplikace měla kopírovat funkcionalitu již běžící webové aplikace.

Výhody mobilních webů

- rychlost vývoje – pouze změna CSS v již běžící webové aplikaci,
- univerzálnost – jedna aplikace pro osobní počítače i mobilní zařízení,
- nevyžaduje instalaci aplikace v telefonu,
- rychlost nasazení aplikace – není potřeba zdlouhavé distribuce přes obchod třetích stran.

Nevýhody mobilních webů

- přístup k funkcím telefonu – zcela chybí napojení na API telefonu,
- chybí nativní UI prvky,
- povinnost být on-line,
- otevírání přes prohlížeč – chybí vlastní ikona aplikace v menu (lze částečně nahradit odkazem na stránku umístěným dle platformy v menu nebo na domácí stránce OS).

2.1.4 Srovnání přístupů k vývoji

Vývoj nativních aplikací pro každou platformu zvlášť byl až do nedávna jediným možným rozumným přístupem k vývoji mobilních aplikací. Bylo to způsobené hlavně nedostatkem alternativ, které by dokázaly nahradit nativní vývoj. Tento problém se

v dnešní době smazává díky stále se zvyšujícímu výkonu mobilních zařízení. Alternativa hybridních aplikací díky tomu nabývá na oblíbenosti a velmi rychle se rozšiřuje. Stejně tak alternativa mobilních webů je v dnešní době takřka nutností. V případě nejjednodušších aplikací dokáže mobilní web tuto aplikaci plnohodnotně nahradit. Ovšem zde je nutností aktivní připojení k internetu. Ve většině případů je tedy pro uživatele přívětivější a pohodlnější samostatná aplikace s vlastní ikonou v menu. Velké firmy tak většinou nabízí kombinaci obou přístupů – kromě mobilního webu nabízí také plnohodnotnou mobilní aplikaci. Dávají tak uživateli možnost výběru a zároveň určitou svobodu. Například uživatel na svém vlastním mobilním zařízení používá plnohodnotnou aplikaci, zároveň se může do aplikace připojit z jakéhokoliv jiného zařízení pomocí mobilního webu.

V Tabulka 1 jsou uvedené vhodné a nevhodné typy aplikací pro všechny výše popsané přístupy k vývoji.

	Vhodné použití	Nevhodné použití
Nativní vývoj	Složité animace 2D/3D hry Aplikace určené pro jednu platformu Aplikace vyžadující méně používaná API	Vývojářsky jednoduché aplikace Prototypování Multiplatformní aplikace
Hybridní vývoj	Aplikace pro stahování dat z API 2D hry Prototypování Aplikace s důrazem na nízkou cenu Multiplatformní aplikace	Výpočetně náročné operace Složité animace 3D hry Rozšířená realita
Mobilní web	Jednoduché aplikace s důrazem na nízkou cenu Aplikace nevyžadující příliš specifické služby telefonu Aplikace, které nemusejí dodržovat guidelines Aplikace, která je založena na webové verzi	Nevhodné například pro načítání kontaktů z telefonu Také ke všemu, co hybridní aplikace

Tabulka 1 – Vhodné a nevhodné typy aplikací pro konkrétní přístupy k vývoji [1]

Většina mobilních aplikací v dnešní době funguje velice podobně na základě manipulace s daty pomocí webového API – načítání API z webu a práce s ním spojená s využitím některých služeb a rozšíření mobilního zařízení. Nejpoužívanějšími službami

jsou typicky získávání údajů z GPS (geolokace), přístup ke kontaktům, použití fotoaparátu v mobilním zařízení, gyroskop, a podobně. Tyto všechny služby dnes hravě zvládají hybridní aplikace. Proto je hybridní vývoj pro tento typ aplikací velmi dobrou volbou.

2.2 Geografické informační systémy v mobilních aplikacích

Geografické informační systémy (GIS) jsou od počátku své existence záležitostí čistě osobních počítačů. S nástupem výkonných mobilních zařízení však GIS přicházejí s různými možnostmi mobilních řešení. Stejně tak jako všechny jiné obory informačních technologií i GIS nabízejí své vlastní mobilní aplikace. V oboru GIS jsou tyto aplikace samozřejmě určené pro práci s mapami a prostorovými daty. Příklady nejen mobilních služeb a aplikací geografického systému ArcGIS jsou uvedeny v [4].

Dnešní GIS však tímto nekončí. Nabízejí také nejrůznější služby starající se o publikaci uživatelských dat – od různých vlastních API pro práci s mapami využitelných v různorodých programovacích jazycích až po komplexní uživatelsky velice přívětivá studia pro vytváření vlastních mobilních mapových aplikací. Tato komplexní studia jsou však určená spíše širší veřejnosti, nabízejí snadný vývoj jednoduché mapové mobilní aplikace. Pro složitější aplikace však již tato studia nejsou vyhovující, stejně jako pro zkušenějšího vývojáře jsou nabízené služby nedostačující. Mezi taková komplexní studio patří například AppStudio for ArcGIS uvedené v [4]. AppStudio bude dále popsáno v kapitole 3.2.8.

Pro využití map v mobilních aplikacích nabízí své služby také většina mapových portálů, jako například Google Maps či OpenStreetMap. Jedná se opět o různá API pro mnoho programovacích jazyků. Vývojář tak získává svobodný přístup k mapě bez různých svazujících vlastností nabízených aplikací GIS, s mapou pracuje ve své vlastní vyvíjené aplikaci.

3 PŘEHLED DOSTUPNÝCH VÝVOJOVÝCH PROSTŘEDÍ

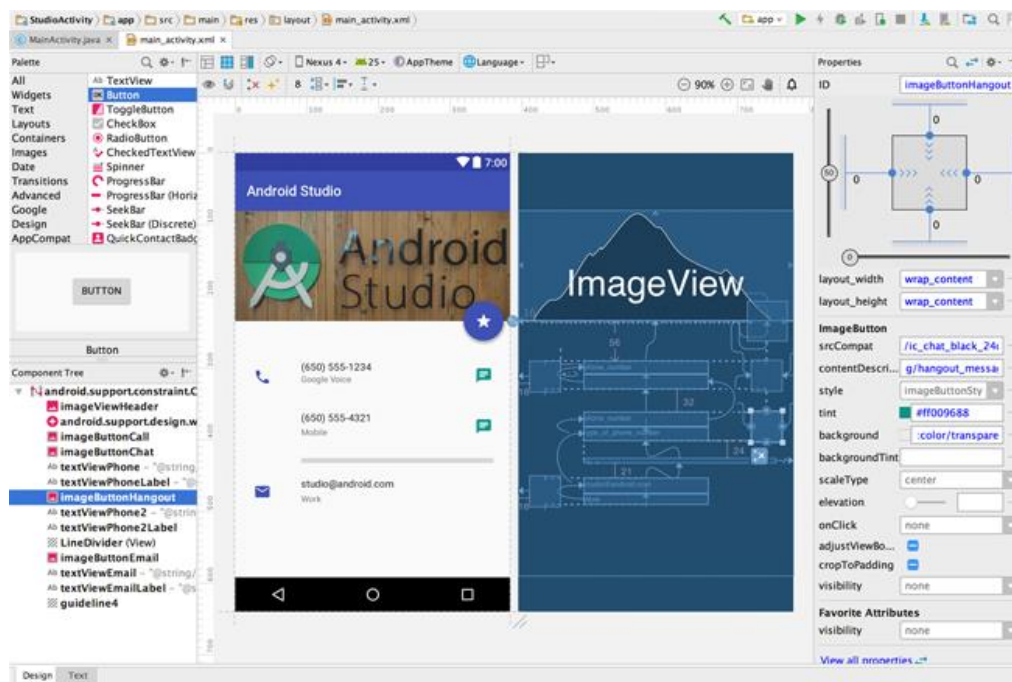
V této kapitole budou představena dostupná vývojová prostředí (IDE) mobilních aplikací. Nejprve budou popsána typická prostředí pro nativní vývoj, poté budou uvedena prostředí pro vývoj multiplatformních aplikací. Mobilním webům nadále nebude věnována pozornost, neboť nejsou předmětem této práce.

3.1 Nativní vývojová prostředí

Mezi nativní vývojová prostředí se řadí prostředí vyvíjená/doporučená přímo vývojáři mobilních platforem. Patří sem Android Studio pro platformu Android od Google, Xcode pro platformu iOS od Apple a Microsoft Visual Studio pro platformu Windows Phone od Microsoft.

3.1.1 Android Studio

Aplikace Android Studio je oficiální IDE pro platformu Android. Nahrazuje dříve doporučený Eclipse s pluginem Android Developer Tools. Poskytuje nástroje pro vytváření aplikací pro každý typ zařízení Android. Nabízí nástroje přizpůsobené vývojářům pro Android, včetně nástrojů pro editaci, ladění, testování a vytváření profilů. Dále obsahuje flexibilní systém sestav a okamžitý systém sestavování a nasazení vyvíjených aplikací. [5]



Obrázek 1 – Ukázka vývojového prostředí Android Studio 2.3 – Design mód [5]

Android studio je volně dostupné ke stažení na vlastním webu [5] a je k dispozici pro uživatele na platformách Windows, Mac OS X i Linux. Je postaveno nad komerčním vývojovým prostředím IntelliJ IDEA. Instalace vývojového prostředí je velmi jednoduchá, nevyžaduje žádné dodatečné instalování a konfiguraci různých pluginů a rozšíření, jak tomu bylo dříve u Eclipse. Součástí instalačního balíčku je:

- samotné Android Studio IDE,
- Android SDK Tools,
- kompilátor Android – flexibilní sestavovací systém Gradle¹,
- základní emulátory systému Android [6].

Aplikace v Android Studiu jsou vyvíjeny v programovacím jazyce Java. To vyžaduje kromě samotného instalačního balíčku Android Studia ještě instalaci JDK (Java Development Kit) od Oracle Corporation. JDK obsahuje soubor základních nástrojů pro vývoj aplikací pro platformu Java.

Jak již bylo zmíněno výše, Android Studio využívá k programování chování aplikace programovací jazyk Java. Design aplikace lze v Android Studiu navrhovat pomocí značkovacího jazyka XML nebo v Design módu (Obrázek 1), což je editor pro navrhování uživatelského rozhraní metodou drag-and-drop. Oproti podobným editorům dříve používaným v Eclipse se Design mód Android Studia vyznačuje velmi vysokou přesností a jednoduchostí vkládání prvků. [6] [7]

3.1.2 Xcode

Xcode je jednotné vývojové prostředí společnosti Apple. Obsahuje balíček profesionálních vývojářských nástrojů pro vývoj aplikací na platformy jak iOS, tak i Mac OS. Pomocí jednoho vývojového prostředí lze tedy vyvíjet aplikace pro všechna zařízení společnosti Apple – počítače Mac, mobilní telefony iPhone, tablety iPad, chytré hodinky Apple Watch a chytré televize Apple TV. [8]

Xcode je nabízený volně ke stažení v Mac App Store², ovšem dostupný pouze pro operační systémy OS X. Pro vývoj aplikací se používá programovací jazyk Swift. Mezi nejvýznamnější vývojářské nástroje v Xcode patří:

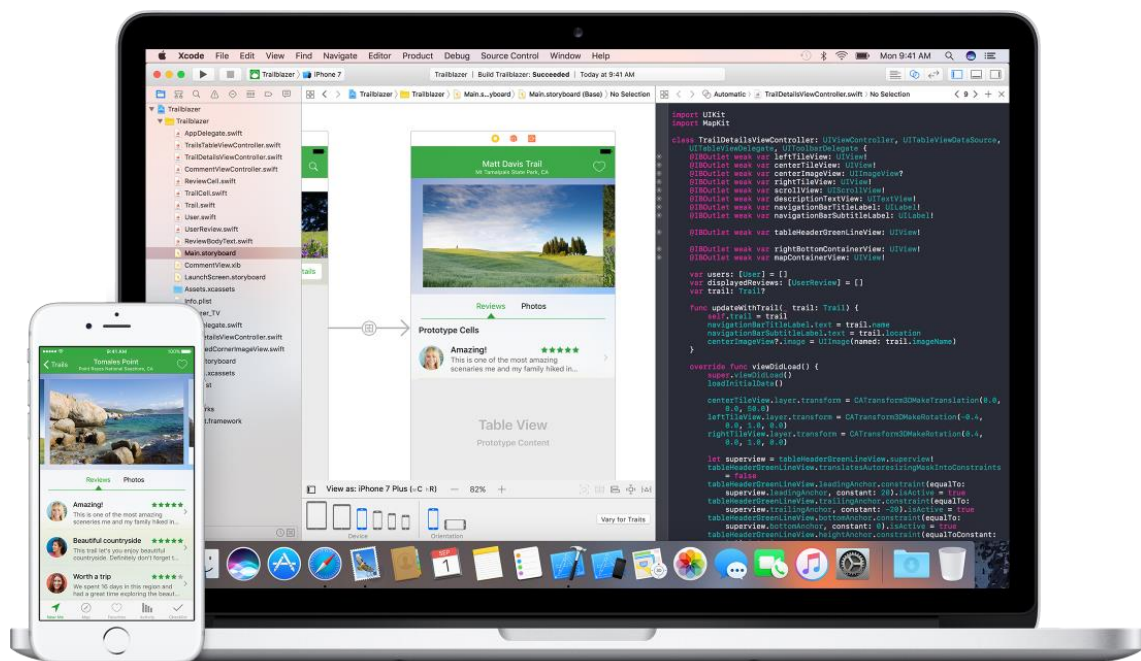
- LLVM kompilátor – založený na open sourceovém projektu LLVM³,
- LLDB debugger⁴ – vyvíjený po boku projektu LLVM,
- Interface Builder – návrhové prostředí pro tvorbu plnohodnotného uživatelského rozhraní bez psaní jakéhokoliv kódu (Obrázek 2),
- iOS Simulator – simulace iPhone/iPad na počítači,
- Instruments – analýza výkonu aplikací.

¹ <https://gradle.org/>

² <https://www.apple.com/uk/osx/apps/app-store/>

³ <http://llvm.org/>

⁴ <https://lldb.llvm.org/>



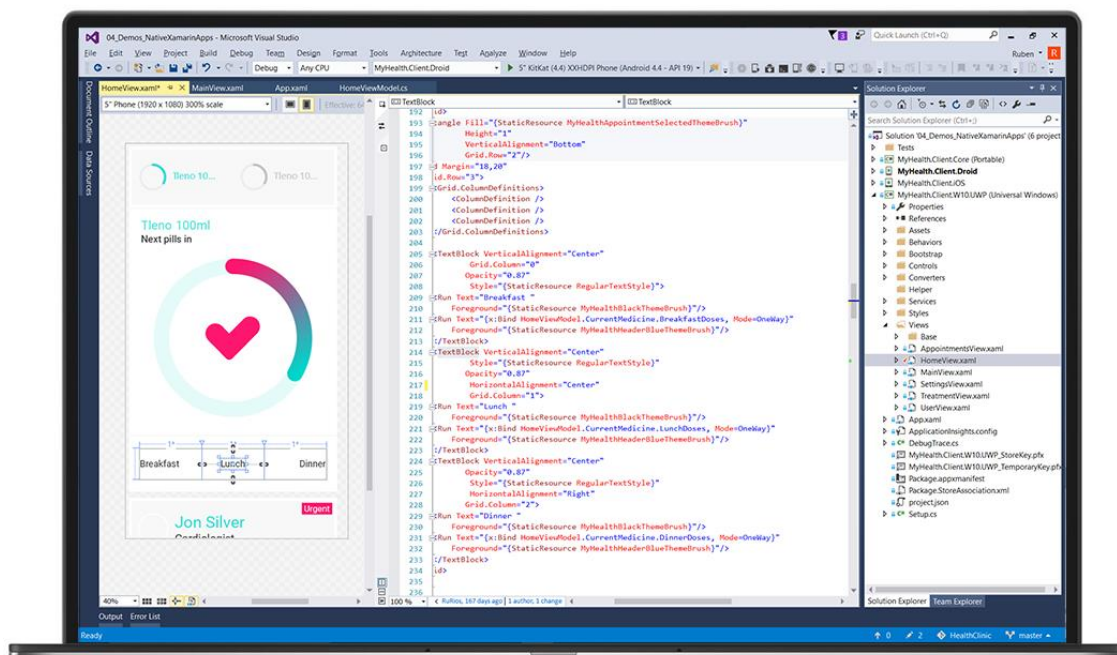
Obrázek 2 – Ukázka vývojového prostředí Xcode – Interface Builder [8]

3.1.3 Visual Studio

Microsoft Visual Studio je vývojové prostředí od Microsoft pro vývoj konzolových aplikací i aplikací s grafickým rozhraním, včetně webových stránek, webových aplikací, mobilních aplikací a mnoho dalšího. Visual Studio je kromě platformy Windows dostupné i pro platformu macOS, v obou případech zdarma pro studenty, projekty open source a samostatné vývojáře. Pro týmové projekty je pak Visual Studio zpoplatněno. Součástí instalačního balíčku poslední verze Visual Studia [9] je:

- nejnovější verze sady Visual Studio, vydaná v březnu 2017 (Obrázek 3),
- nejnovější sada Windows Standalone SDK (verze 10.0.15063.0),
- šablony aplikací pro Univerzální platformu Windows (UPW) pro tablety, telefony, počítače, Windows IoT nebo Xbox,
- šablony desktopových aplikací pro Windows,
- emulátory systému Windows 10 Mobile.

Microsoft se v posledních verzích Visual Studia na rozdíl od výše uvedených vývojových prostředích vydává ve vývoji mobilních aplikací cestou multiplatformní. Vývoj multiplatformních aplikací zajišťuje ve Visual Studiu vývojová platforma Xamarin (více v kapitole 3.2.1). Základní platformou, pro kterou se aplikace vyvíjejí, je samozřejmě Windows (Windows Phone, Windows 10 Mobile). Pro vývoj se používá programovací jazyk C#. Díky používání stejného jazyka, API a datových struktur je však možné pomocí sdílených projektů a přenosných knihoven tříd sdílet okolo 75 % aplikačního kódu na platformách iOS a Android. [9]



Obrázek 3 – Ukázka vývojového prostředí Visual Studio 2017 [9]

3.2 Multiplatformní (hybridní) vývojová prostředí

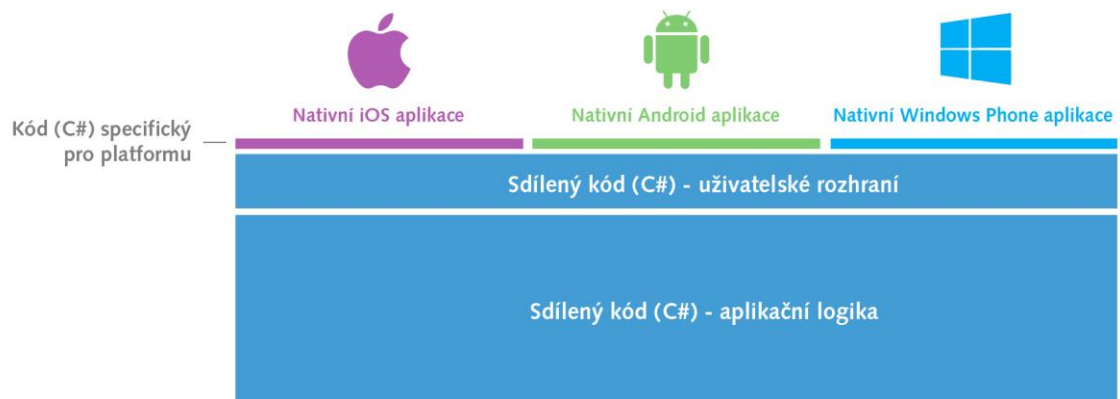
Hybridní vývojová prostředí jsou prostředí zajišťující využití stejného aplikačního kódu napříč platformami. V následujících podkapitolách budou představena vývojová prostředí založená na sdílení aplikačního kódu mezi platformami, hybridní vývojová prostředí založená na JavaScriptu a komplexní studia pro snadné vytváření jednoduchých mobilních aplikací.

3.2.1 Xamarin

Xamarin [10] je vývojová platforma pro vytváření nativních aplikací pro 3 hlavní mobilní operační systémy: iOS, Android a Windows Phone. Architektura Xamarinu se snaží maximalizovat opakované použití jednou napsaného kódu, který je sdílen mezi všechny požadované platformy. Množství takto sdíleného kódu se v běžných aplikacích pohybuje okolo 75 %. Xamarin ale není jen platforma "jednou zapsat, spustit všude". Jednou z jejích silných stránek je schopnost implementovat nativní uživatelská rozhraní specificky pro každou platformu. Na Obrázek 4 je poměrově znázorněno množství sdíleného kódu a kódu specifického pro danou platformu. Díky sdílení kódu mezi platformami zaujímá Xamarin místo mezi multiplatformními vývojovými prostředími. Nejedná se však o vývojové prostředí hybridních aplikací, neboť výstupem kompilace Xamarinu jsou nativní aplikace pro dané platformy (viz. Dále).

Xamarin pro vývoj aplikací používá jediný programovací jazyk, a to C#. Tento jazyk je nativním jazykem pro vývoj aplikací pro platformu Windows, tedy pro vývoj mobilních aplikací pro Windows není třeba žádných dalších speciálních nástrojů. Pro vývoj aplikací pro zbylé platformy jsou součástí Xamarinu knihovny MonoTouch.dll pro

iOS a MonoAndroid.dll pro Android. Tyto knihovny obsahují mimo různé sestavy a třídy pro dané platformy také obálky pro téměř celý iOS SDK a Android SDK, které umožňují vyvolání základních rozhraní API přímo z kódu v C#. [11] [12]



Obrázek 4 – Sdílení kódu v Xamarinu [10]

Kompilace do platforem iOS a Android

Xamarin nabízí dva komerční produkty: Xamarin.iOS a Xamarin.Android (pro iOS a Android). Pomocí těchto dvou produktů prostředí Xamarin kompiluje napsaný kód v C# do nativního strojového kódu příslušné platformy. V případě iOS probíhá tato kompilace přímo. V systému Android probíhá kompilace do mezikódu Intermediate Language (IL), který je v okamžiku spuštění aplikace kompilován just-in-time (JIT) kompilátorem do nativního strojového kódu pro daný procesor. [12]

Výsledkem kompilace aplikací v Xamarinu je aplikační balíček, buď soubor „.app“ v systému iOS nebo soubor „.apk“ v systému Android. Tyto soubory jsou nerozlišitelné od balíčků aplikací vytvořených ve výchozích IDE platformách a jsou publikovatelné stejným způsobem. [12]

3.2.2 Apache Cordova

Apache Cordova je open-source vývojový framework pro vytváření mobilních aplikací. Patří mezi typické představitele hybridních vývojových prostředí. Umožňuje použití standardních webových technologií – HTML5, CSS3 a JavaScript pro vývoj na různých platformách. Aplikace se spouštějí v obalech (WebView) cílených na každou platformu a využívají standardizované API pro přístup ke všem možnostem nativního zařízení, jako jsou senzory, data, stav sítě atd. Pomocí Apache Cordova lze rozšířit aplikace na více platforem, aniž by musely být znovu implementovány s jazykem a sadou nástrojů každé platformy. [13]

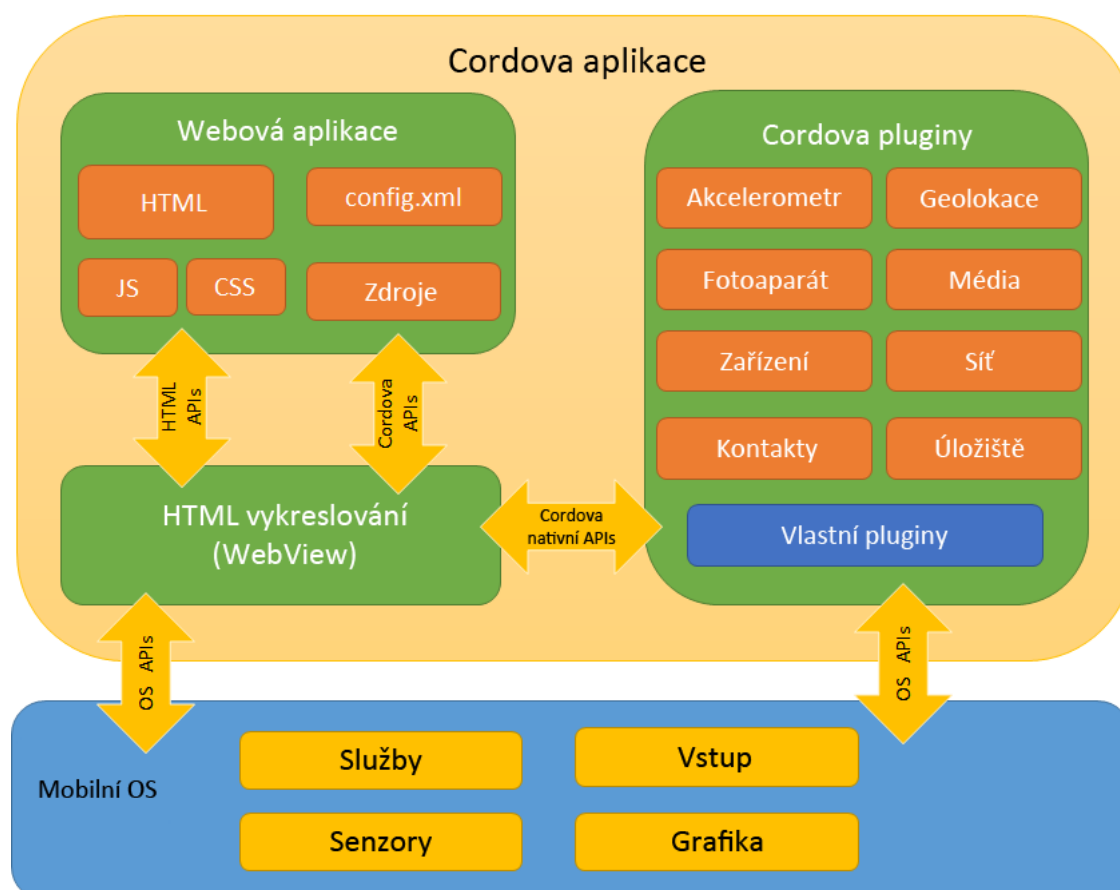
Apache Cordova je dostupný v různé míře pro platformy Android, BlackBerry, iOS, Ubuntu, Windows Phone 8, Windows (8.1, 10) a OS X. Přesné informace o dostupnosti jednotlivých součástí Cordovy na jednotlivých platformách jsou přehledně uvedeny v Tabulka 2.

	Android	iOS	Windows Phone 8	Windows (8.1, 10)	BlackBerry 10	Ubuntu	OS X
Cordova CLI	Ano	Ano	Ano	Ano	Ano	Ano	Ano
Vestavěný WebView	Ano	Ano	Ne	Ne	Ne	Ano	Ano
Rozhraní pro pluginy	Ano	Ano	Ano	Ano	Ano	Ano	Ano

Tabulka 2 – Apache Cordova – podporované platformy [14]

Architektura

K aplikacím Apache Cordova existuje několik komponent. Obrázek 5 zobrazuje schéma architektury aplikace vytvořené ve frameworku Apache Cordova.



Obrázek 5 – Schéma architektury Cordova aplikace [13]

Webová aplikace je hlavní část architektury obsahující kód aplikace. Samotná aplikace je implementována jako webová stránka. Základem je místní soubor s názvem

index.html, který odkazuje na CSS, JavaScript (JS), obrázky a další zdroje. Součástí Webové aplikace je také soubor config.xml, který poskytuje informace o aplikaci a určuje parametry ovlivňující její chování. Webová aplikace se spouští v aplikaci WebView v rámci nativního aplikačního balíčku, který je konečným výstupem sestavení aplikace.

WebView společně s funkcemi Cordovy poskytuje aplikace s kompletním uživatelským rozhraním. Na některých platformách může být WebView součástí větší hybridní aplikace, která kombinuje WebView s komponentami nativní aplikace.

Cordova pluginy jsou velmi důležitou součástí celého systému. Poskytují rozhraní pro vzájemnou komunikaci mezi Cordovou a nativními komponentami a navazují přístup k standardním API daného zařízení. To umožňuje volat nativní kód z JavaScriptu. Apache Cordova nabízí širokou sadu pluginů nazvanou „Core Plugins“. Pluginy z této sady poskytují aplikaci přístup k možnostem zařízení, jako je akcelerometr, geolokace, fotoaparát a další. Přehled všech pluginů „Core Plugins“ s dostupností pro jednotlivé podporované platformy je přehledně uveden v Tabulka 3. Kromě „Core Plugins“ existují i pluginy třetích stran, které poskytují vazby na méně obvyklé funkce zařízení. Tyto funkce však nemusejí nutně být k dispozici na všech platformách. Apache Cordova umožňuje také vývoj vlastních pluginů, které mohou být nutné například pro komunikaci mezi Cordovou a vlastními nativními komponentami. Vytvořený nový projekt v Cordově nemá v základu žádné pluginy. Všechny požadované pluginy je nutno explicitně přidat. Cordova neposkytuje žádné UI prvky, zajišťuje pouze chod aplikace. [13]

Vytvoření mobilní aplikace

Cordova poskytuje dva základní pracovní postupy pro vytvoření mobilní aplikace:

- pracovní postup pro více platform (CLI),
- pracovní postup zaměřený na platformu.

Pracovní postup pro více platform se používá pro vývoj aplikací fungujících na co nejvíce různých mobilních operačních systémech, přičemž není potřeba konkrétního vývoje na jednotlivé platformy. Základem tohoto postupu je nástroj vyšší úrovně – Cordova CLI. Ten umožňuje vytvářet projekty pro více platform najednou. CLI kopíruje společnou sadu webových prostředků do podadresářů pro každou mobilní platformu, provádí změny konfigurace pro každou platformu a vytváří binární soubory aplikací. Dále také služba CLI nabízí společné rozhraní pro aplikování pluginů do aplikace. Tento postup je také doporučený vývojáři Cordovy.

Pracovní postup zaměřený na platformu se používá pro vývoj aplikací určených pro jednu platformu v případě, kdy vývojář potřebuje aplikaci měnit na nižší úrovni. Tento postup je nutné použít například při kombinaci vlastních nativních součástí s webovými komponentami Cordovy. Základem tohoto postupu je sada skriptů, které jsou přizpůsobeny pro každou podporovanou platformu, a samostatný nástroj Plugman, který umožňuje aplikovat pluginy. Pomocí tohoto postupu lze vyvíjet aplikace i pro více platform. Je to ale obecně obtížnější z důvodu chybějícího vyššího nástroje. V takovém případě by bylo nutné každou platformu řídit zvlášť a také zvlášť modifikovat pluginy pro každou platformu.

Podle zvoleného pracovního postupu se také liší vlastní instalace Cordovy. Je tedy nutné si vhodný pracovní postup zvolit již před instalací frameworku. [13]

	Android	iOS	Windows Phone 8	Windows (8.1, 10)	BlackBerry 10	Ubuntu	OS X
Akcelerometr	Ano	Ano	Ano	Ano	Ano	Ano	Ne
Status baterie	Ano	Ano	Ano	Ano	Ano	Ano	Ne
Kamera	Ano	Ano	Ano	Ano	Ano	Ano	Ne
Fotoaparát	Ano	Ano	Ano	Ano	Ano	Ano	Ne
Kompas	Ano	Ano	Ano	Ano	Ano	Ano	Ne
Připojení	Ano	Ano	Ano	Ano	Ano	Ano	Ne
Kontakty	Ano	Ano	Ano	část.	Ano	část.	Ne
Zařízení	Ano	Ano	Ano	Ano	Ano	Ano	Ano
Události	Ano	Ano	Ano	Ano	Ano	Ano	Ne
Soubory	Ano	Ano	Ano	Ano	Ano	Ano	Ano
Přenos souborů	Ano	Ano	Ano	Ano	Ano	Ne	Ne
Geolokace	Ano	Ano	Ano	Ano	Ano	Ano	Ne
Globalizace	Ano	Ano	Ano	Ano	Ano	Ano	Ne
InApp Prohlížeč	Ano	Ano	Ano	část.	Ano	Ano	Ne
Média	Ano	Ano	Ano	Ano	Ano	Ano	Ne
Notifikace	Ano	Ano	Ano	Ano	Ano	Ano	Ne
Úvodní obrazovka	Ano	Ano	Ano	Ano	Ano	Ano	Ne
Stavový řádek	Ano	Ano	Ano	Ano	Ne	Ne	Ne
Úložiště	Ano	Ano	Ano	Ano	Ano	Ano	Ne
Vibrace	Ano	Ano	Ano	Ano	Ano	Ano	Ne

Tabulka 3 – Apache Cordova – Core plugins [14]

3.2.3 PhoneGap

PhoneGap společnosti Adobe je distribuce vývojového frameworku Apache Cordova, který je hlavním prostředkem pohánějícím PhoneGap. Framework původně vznikl jako PhoneGap ve společnosti Adobe. Pro udržení open-source licence a dodržení standardů byl poté PhoneGap předán společnosti Apache, kde se jméno frameworku změnilo na Cordova.

PhoneGap je tedy stejný multiplatformní framework jako Apache Cordova a nabízí totožné nástroje a možnosti. Oproti Cordově však navíc nabízí některé další služby společnosti Adobe, které však již mohou být zpoplatněny. Příkladem takové placené služby společnosti Adobe je služba PhoneGap Build Service. Jde o službu umožňující vytvářet aplikace v cloudu bez nutnosti místní instalace SDK. [15] [16] [17]

3.2.4 Ionic

Ionic je kompletní open-source SDK pro vývoj hybridních mobilních aplikací pomocí webových technologií HTML, CSS a JavaScript. Oproti jiným hybridním SDK se Ionic zaměřuje na vzhled aplikace, uživatelský pocit z ní a na interaktivitu uživatelského rozhraní. Ionic SDK je dostupné pro operační systémy Mac OS X, Linux a Windows. Doporučeným operačním systémem pro vývoj je Mac OS X, protože v ostatních OS nelze překládat aplikace pro platformu iOS. Ionic SDK se skládá z velkého množství různých nástrojů a služeb. Mezi ty nejvýznamnější patří:

- Ionic framework,
- rozhraní CLI – výkonné rozhraní příkazového řádku,
- Ionic komponenty a témata,
- Icons,
- Ionic Creator.

V následujících odstavcích je čerpáno převážně z dokumentace Ionicu [18], ale také z článků [19] [20] [21].

Ionic framework

Nejdůležitější částí Ionic SDK je Ionic framework, který je vytvořený na bázi javascriptového webového frameworku AngularJS⁵. Pro sestavování aplikací na cílových platformách je používán framework Apache Cordova. V Ionic frameworku jsou tedy přístupné všechny služby Cordovy, tzn. hlavně zobrazení webové aplikace v komponentě WebView a poskytnutí spojení s nativními službami (API) v mobilním zařízení.

První alfa verze Ionic frameworku byla vydána v listopadu 2013. V květnu 2015 byla dokončena finální verze 1.0 a započal vývoj verze 2.0. Ten byl dokončen na konci ledna 2017.

⁵ <https://angularjs.org/>

Verze 1.0 podporuje platformy iOS a Android ve verzích operačních systémů iOS 7 a vyšší a Android 4.1 a vyšší. Platforma Windows není v Ionic verzi 1.0 oficiálně podporována, avšak existují postupy, jak aplikace zpřístupnit i pro tuto platformu. Pro zlepšení výkonu na starších zařízeních je doporučeno použití nástroje Crosswalk⁶, což je speciální WebView komponenta, která je přibalená do samotné aplikace.

Ve verzi 2.0 přibyla již oficiální podpora platformy Windows. Jsou podporovány operační systémy iOS 8 a vyšší, Windows 10 a Android 4.4 a vyšší (podporu lze vrátit zpět na Android 4.1 použitím nástroje Crosswalk).

Rozhraní CLI – výkonné rozhraní příkazového řádku

Rozhraní CLI je velmi výkonné rozhraní příkazového řádku. Poskytuje řadu užitečných příkazů pro vývoj Ionic aplikací. Kromě samotné instalace a aktualizace Ionicu obsahuje CLI vestavěný webový server, nástroje pro sestavování a ladění aplikací a mnoho dalšího.

Ionic komponenty a témata

Za komponenty jsou v Ionicu označovány opakovaně použitelné prvky UI, které slouží jako stavební prvky pro mobilní aplikaci. Komponenty jsou tvořeny pomocí HTML a CSS, v některých případech i s přispěním JavaScriptu. Každá Ionic komponenta podléhá tzv. platformní kontinuitě. To znamená, že se každá Ionic komponenta přizpůsobuje platformě, na které je aplikace spuštěna. Ionic ve verzi 2.0 nabízí opravdu velké množství komponent – tlačítka, upozornění, gesta, seznamy, menu, záložky a mnoho dalšího⁷.

Témata jsou sady stylů CSS, které se na aplikaci aplikují. Ionic standardně používá světlé téma, ale je možné využít i téma tmavé či předpřipravená témata jiných barev. Pro použití vlastního barevného tématu stačí na jednom místě kódu nastavit novou hodnotu základní barvy základního světlého tématu. Ionic v základu použije světlé téma s definovanou novou hodnotou základní barvy. Všechny prvky UI, včetně všech dalších barevných odstínů tématu, různých barevných přechodů apod., budou automaticky přepočítány z hodnoty základní barvy. Vlastní barevné téma je tím tedy kompletně hotové. Pro toto globální nastavení se v Ionicu využívají proměnné SASS, které umožňují definovat hodnotu a použít ji na více místech jako vlastnosti CSS. Všechny vlastnosti CSS lze poté pozměnit pomocí vlastních stylů CSS.

Kromě témat umožňuje platformní kontinuita přiřadit komponentům styly specifické pro jednotlivé platformy. Styly celé aplikace se tedy budou měnit na základě platformy, na které se aplikace spouští. To umožňuje uživatelům nabídnout uživatelskou zkušenost, na kterou jsou na svém mobilním zařízení zvyklí.

Icons

Ionicons⁸ je prémiová open-source sada ikon pro Ionic Framework. Patří mezi základní součásti Ionic SDK. Použití ikon v aplikaci se provádí definováním CSS třídy s názvem ikony libovolnému HTML prvku.

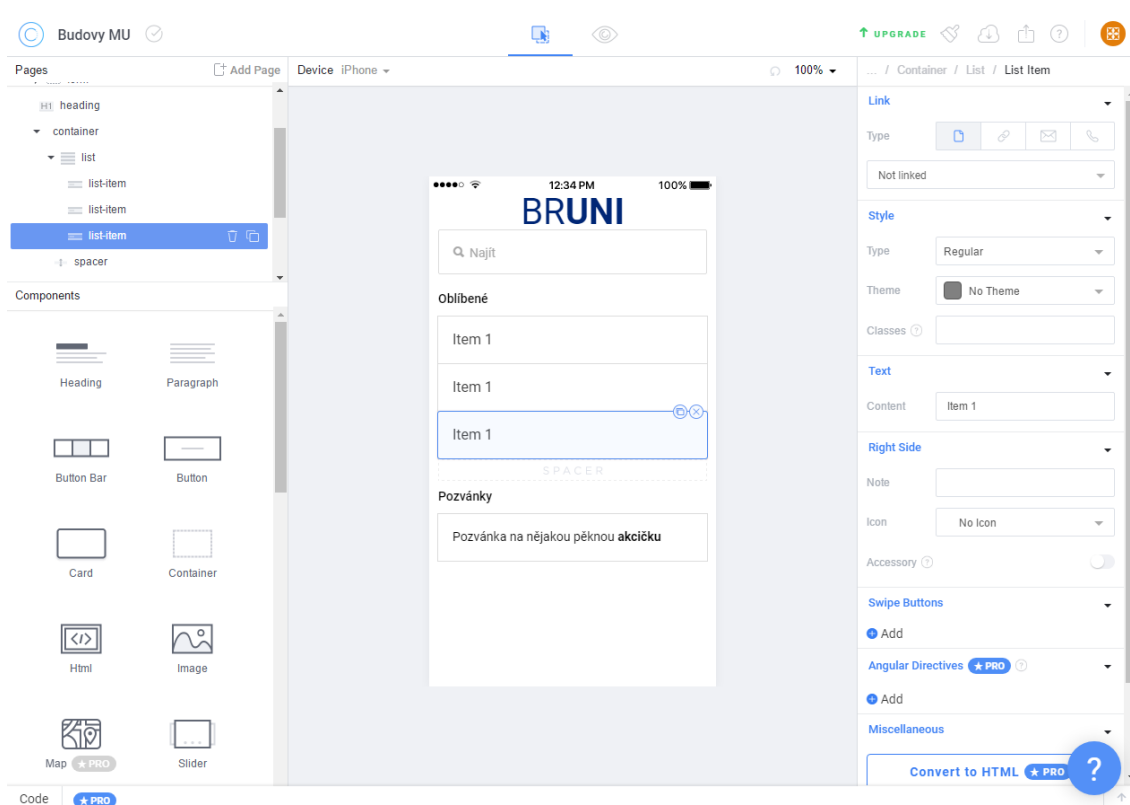
⁶ <https://crosswalk-project.org/>

⁷ <https://ionicframework.com/docs/components/#overview>

⁸ <http://ionicons.com/>

Ionic Creator

Ionic Creator je jednoduchý webový nástroj pro rychlejší vytváření mobilních aplikací pouze pomocí přetahování prvků myši. V základní volně přístupné verzi umožňuje vytvořit jeden projekt. V rámci projektu si pak vývojář připraví jednotlivé obrazovky zamýšlené aplikace, kterou si posléze může vyexportovat a předpřipravené obrazovky tak využít při dalším vývoji. Příklad takové předpřipravené obrazovky v Ionic Creator je na Obrázek 6. Ve zpoplatněné plné verzi Ionic Creator je kromě libovolného množství projektů také umožněno doplnit logiku aplikace a aplikaci následně sestavit přímo do aplikačního balíčku.



Obrázek 6 – Ionic Creator

Shrnutí

Ionic se během posledních pár let stal jedním z nejoblíbenějších prostředků pro vývoj hybridních mobilních aplikací. Nyní má masivní mezinárodní komunitu vývojářů, kteří přispívají k rychlému růstu a adaptaci Ionicu. V roce 2015 bylo Ionic vývojáři vytvořeno více než 1,3 milionu mobilních aplikací.

3.2.5 React Native

React Native je dalším představitelem multiplatformního vývojového prostředí, které nevytváří hybridní aplikace, ale plně nativní aplikace podobně jako Xamarin. Pro vývoj aplikací je využit pouze JavaScript s využitím javascriptové knihovny pro vytváření

uživatelských rozhraní React⁹. React Native používá stejný design jako knihovna React, jen namísto webových komponent využívá nativní komponenty. Jsou používány stejné základní bloky uživatelského rozhraní jako v běžných nativních aplikacích platforem iOS a Android. Jen na rozdíl od nativního vývoje jsou tyto bloky seskupovány pomocí JavaScriptu a knihovny React. Výsledná aplikace je nerozlišitelná od nativní aplikace sestavené pomocí nativního vývojového prostředí.

Vlastnosti React Native

- React – framework pro vytváření webových a mobilních aplikací s použitím JavaScriptu,
 - Native – využití nativních komponent ovládaných JavaScriptem,
 - platformy – podpora platform iOS a Android.
 - Výhody:
 - JavaScript – využití znalostí JavaScriptu z webových aplikací pro vytváření nativních mobilních aplikací,
 - sdílení kódu – použití shodného kódu pro různé platformy,
 - velká komunita vývojářů.
 - Nevýhody:
 - nativní komponenty – knihovna nativních komponentů není tak rozsáhlá jako např. u Apache Cordova; vytváření vlastní nativní funkcionality pouze přes specifický kód platforem (Swift, Java).
- [22] [23]

3.2.6 Appcelerator Titanium

Appcelerator Titanium je open-source framework umožňující vývoj nativních mobilních aplikací pro platformy iOS, Android a Windows. Využívá k tomu jednoduchou javascriptovou kódovou základnu vyvinutou firmou Appcelerator. Firma Appcelerator také přináší vlastní vývojové prostředí Appcelerator Studio IDE.

Vlastnosti Appcelerator Titanium

- sdílení kódu – 60–90 % napříč platformami,
- Hyperloop – přímý přístup z JavaScriptu k nativním rozhraním API; nabízí více než 5 000 rozhraní API,
- App Designer – vizuální drag-and-drop vývoj aplikací,
- API Builder – rychlé sestavování API, modelů a konektorů pro přístup k datům,
- API Runtime – infrastruktura pro spouštění aplikací,
- platformy – podpora platform iOS, Android a Windows.

⁹ <https://facebook.github.io/react/>

Základní balíček Appcelerator Titanium je zdarma a je určen pro individuální vývojáře. Obsahuje Titanium a Hyperloop, Appcelerator Studio, API Builder a další služby. Placený balíček je určen pro týmové vývojáře a obsahuje navíc především App Designer a nástroje na sdílení a testování beta verzí aplikací. [24]

3.2.7 Sencha Ext JS

Sencha Ext JS je placený framework pro vývoj multiplatformních mobilních webových aplikací založený na HTML 5 a JavaScriptu. Podporovanými platformami jsou všechny hlavní mobilní platformy – iOS, Android, BlackBerry, Windows a další. Tento framework je představitelem aplikací s architekturou typu MVC (model-view-controller). Aplikace je rozdělena do tří nezávislých komponent – datový model, uživatelské rozhraní a řídicí logika. Modifikace jakékoliv komponenty má minimální vliv na ostatní komponenty. Framework využívá techniky hardwarové akcelerace, které poskytují vysoce výkonné komponenty uživatelského rozhraní pro mobilní zařízení. Součástí balíčku Sencha Ext JS je spousta nástrojů, jako například:

- Sencha CMD – plná správa životního cyklu během vývoje aplikace,
- Sencha IDE plugins – nástroje pro usnadnění vývoje aplikací v Ext JS,
- Architect – vizuální drag-and-drop nástroj pro vývoj aplikací,
- Themer – vytváření vlastních aplikačních motivů bez psaní kódu,
- Inspector – nástroj pro ladění s přímým přístupem ke komponentům,
- Stencils – kompletní sada UI prvků. [25]

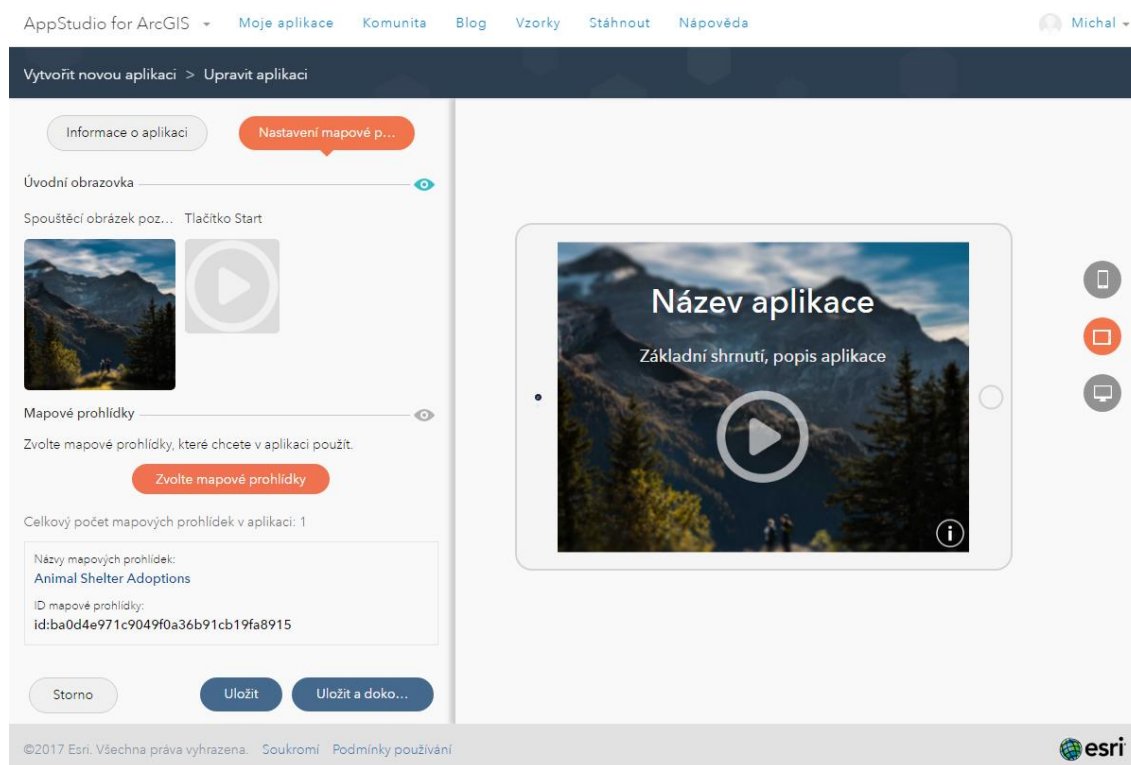
3.2.8 AppStudio for ArcGIS

AppStudio for ArcGIS je vývojové prostředí hybridních aplikací na platformy Android, iOS, Windows, Linux a OS X vyvinuté v rámci geografického informačního systému ArcGIS firmy Esri. AppStudio je postavené tak, že lze aplikace vytvářet i bez znalosti programování. K dispozici jsou různé šablony aplikací, které lze do určité míry upravovat. Se znalostí programování je pak možné vyvíjet vlastní aplikace s využitím zdrojových kódů předpřipravených šablon aplikací. K tvorbě aplikací pomocí AppStudia for ArcGIS je zapotřebí mít založený účet organizace na ArcGIS Online. Aplikace jsou předurčené k používání webových map z ArcGIS Online a dalších služeb geografického informačního systému ArcGIS. [26] [27]

AppStudio je založeno na multiplatformní knihovně Qt a ArcGIS Runtime SDK for Qt. Vývoj aplikací probíhá v deklarativním jazyku QML založeném na JavaScriptu.

Součásti AppStudio for ArcGIS

- AppStudio for ArcGIS Web Edition (Obrázek 7)
 - webové prostředí pro vytváření aplikací z připravených šablon.
- AppStudio for ArcGIS Desktop Edition
 - balík nástrojů pro vývoj vlastních aplikací,
 - obsahuje AppStudio, vývojové prostředí Qt Creator s QML doplňky Esri, šablony aplikací a ukázky kódu,
 - dostupné pro operační systém Windows, Ubuntu a Mac OS X.
- AppStudio Player
 - prostředí pro testování aplikací na cílovém zařízení před kompilací instalačních souborů.
- Webová služba Cloud Make
 - kompilace zdrojového kódu ve webové službě, vrací hotové instalační soubory.



Obrázek 7 – Ukázka prostředí AppStudio for ArcGIS Web Edition

Šablony aplikací

AppStudio nabízí v současnosti tři hotové šablony¹⁰. S jejich pomocí lze vytvořit aplikace zcela bez psaní kódu. Jedná se o šablony Prohlídka mapy, Prohlížeč map a Rychlé hlášení.

¹⁰ <http://appstudio.arcgis.com/create.html>

Šablona Prohlídka mapy slouží k vytvoření aplikace zobrazující mapu s body zájmu a jejich doprovodnými informacemi s důrazem na fotografie.

Šablona Prohlížeč map vytváří aplikaci pro prohlížení webových map. Nabízí základní uživatelské rozhraní pro práci s mapou, dále pak další služby jako vyhledávání, legendu a záložky.

Pomocí šablony Rychlé hlášení lze vytvořit aplikaci pro sběr geografických dat.

Shrnutí

AppStudio for ArcGIS je koncipováno pro vývoj aplikací pracujících s daty geografického informačního systému ArcGIS. Proto je AppStudio pro tento typ aplikací jistě velmi vhodnou volbou. Pro programování obecných aplikací však toto prostředí nemusí poskytovat požadované nástroje a služby.

4 NÁVRH MOBILNÍ APLIKACE

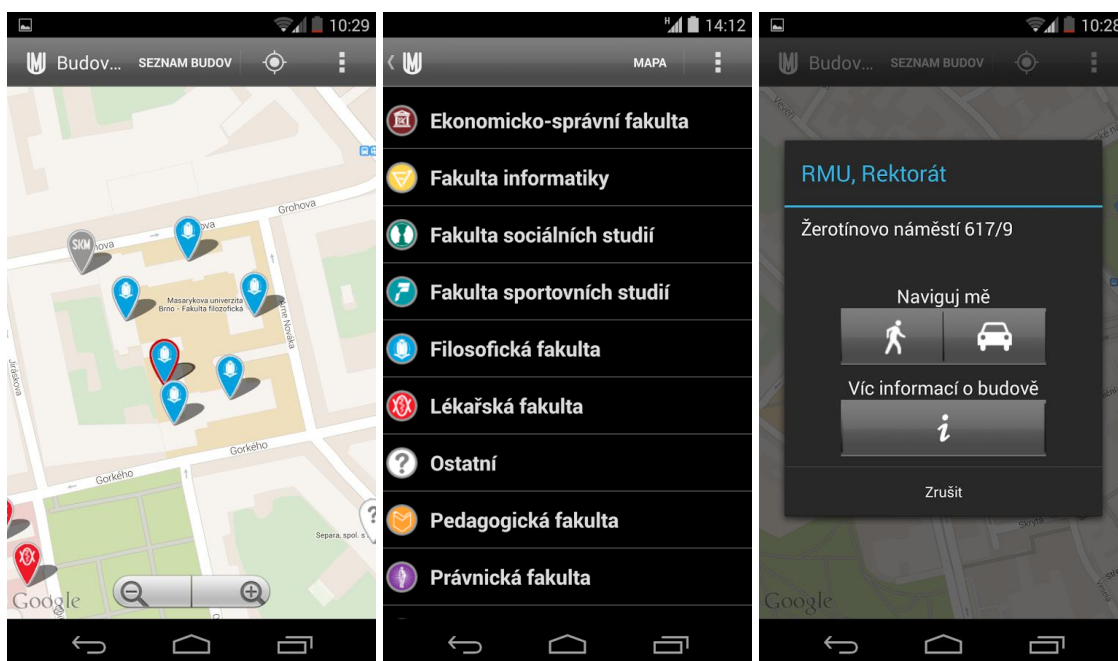
V této kapitole bude provedena analýza a návrh mobilní aplikace pro usnadnění orientace v prostředí velké univerzity. Nejprve však budou představeny již existující aplikace, které řeší stejný, či podobný úkol.

4.1 Konkurenční aplikace

4.1.1 Budovy MU

Aplikace Budovy MU¹¹ (Obrázek 8) je první vyvinutou aplikací zobrazující přehled o budovách Masarykovy Univerzity (MU). Pro Oddělení facility managementu MU ji v roce 2012 vyvinul Martin Vytrhlík. Jedná se o nativní aplikaci pro operační systém Android.

Aplikace nabízí tři základní obrazovky – mapu, seznam budov a nastavení. Po spuštění aplikace se jako úvodní obrazovka zobrazí přímo mapa přiblížená na město Brno se zobrazenými bodovými ikonami budov. V horní části obrazovky je lišta obsahující název/logo aplikace a ovládací tlačítka, pomocí kterých se lze v každém stavu aplikace přepnout do jedné ze základních obrazovek.



Obrázek 8 – Aplikace Budovy MU

Seznam budov obsahuje skupiny budov rozdělených podle příslušnosti k fakultám či jiným organizačním jednotkám MU. Každá skupina má vlastní ikonu. Po kliknutí na skupinu budov se zobrazí výpis budov dané skupiny. Po kliknutí na budovu se aplikace

¹¹ <https://play.google.com/store/apps/details?id=cz.muni.ics.pasport.OPBMapApp&hl=cs>

přesměruje na úvodní obrazovku s mapou, ve které je barevným obrysem zvýrazněna ikona vybrané budovy.

Pro zobrazení údajů o jakékoliv budově je zapotřebí kliknout na její ikonu. Zobrazí se vyskakovací okno, ve kterém lze zvolit mezi spuštěním navigace a zobrazením informací o budově. Informace o budově se zobrazí na samostatné obrazovce. Informace obsahují název budovy, adresu, webovou adresu skupiny, do které budova patří a poté popis budovy obsahující ve většině případů výpis pracovišť. Po spuštění navigace se spustí výchozí navigační aplikace používaná v daném zařízení s označeným cílem pomocí souřadnic vybrané budovy.

V nastavení aplikace je možné určit, které skupiny budov se mají v aplikaci zobrazovat. Dále je možné přepínat podkladovou mapu mezi obecnou a satelitní mapou.

Zobrazovaná data jsou uložena přímo v aplikaci bez možnosti jejich aktualizace. Při aktualizaci dat je nutné aktualizovat celou aplikaci a vydat novou verzi přes obchod Google Play.

Shrnutí

Aplikace je již staršího data výroby a tomu odpovídá i její design. Z uživatelského hlediska je velmi nešikovně vyřešeno získávání informací o budově, kdy je nutné kliknout na její ikonu. Výběr budovy probíhá ve většině případů ze seznamu budov, kdy je uživatel přesměrován na mapu se zvýrazněnou ikonou požadované budovy. Uživatel musí kliknout na tuto ikonu, což ve většině případů znamená nutnost výrazného přiblížení mapy tak, aby byl umožněn jistý výběr požadované budovy. Poté je nutné kliknout ještě na tlačítko informací. To znamená spoustu zbytečných kroků navíc.

Výrazným problémem v aplikaci je také její aktuálnost. Zobrazované informace o budovách se mohou často měnit (např. přesuny pracovišť při rekonstrukcích) a proto je velkým nedostatkem nemožnost aktualizace samotných dat bez nutnosti aktualizace celé aplikace.

4.1.2 MUNI Buildings

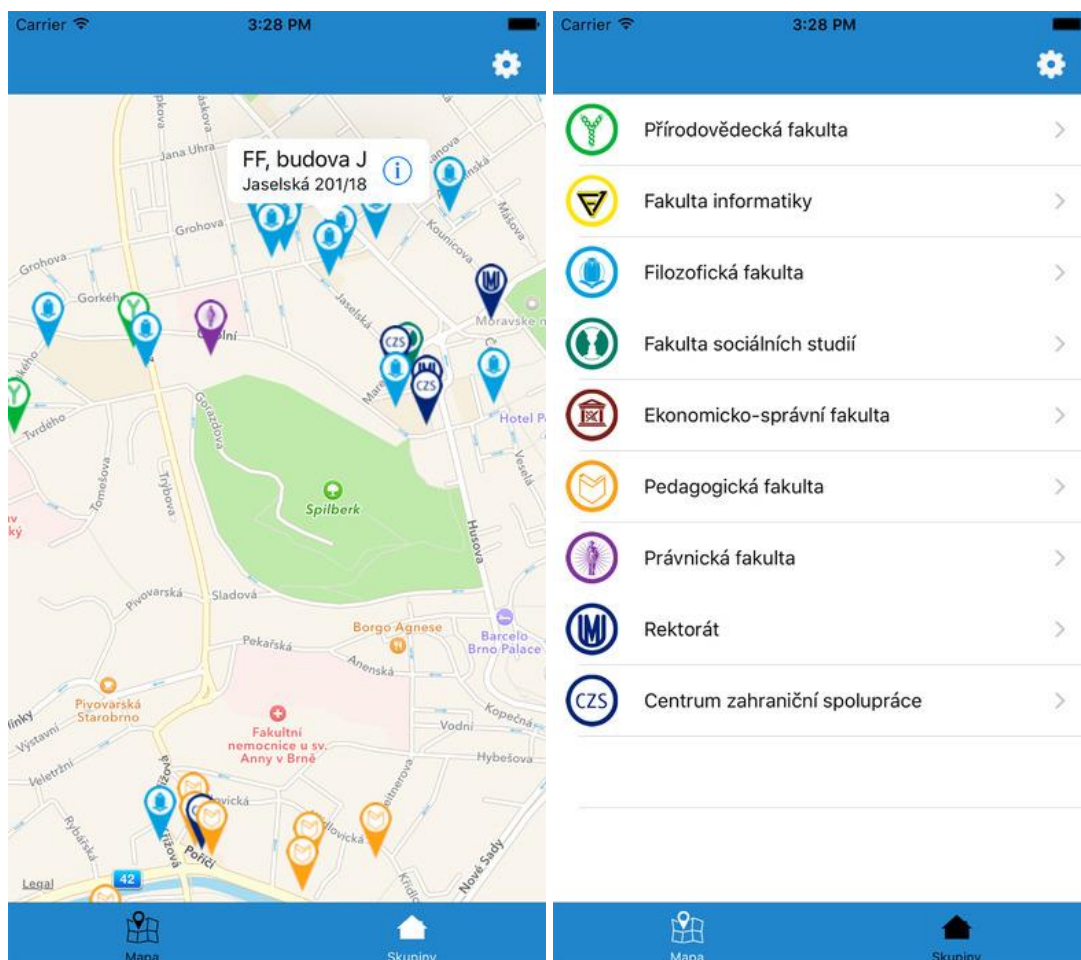
Aplikace MUNI Buildings¹² (Obrázek 9) je klonem aplikace Budovy MU pro operační systémy iOS. Jedná se o nativní iOS aplikaci implementovanou v jazyce Swift. Pro Oddělení facility managementu MU ji implementoval v letech 2015–2016 opět Martin Vytrhlík.

Základ aplikace je velmi podobný aplikaci Budovy MU. Opět nabízí tři základní obrazovky – Mapa, Skupiny a Nastavení. Úvodní obrazovkou je opět mapa přiblížená na město Brno se zobrazenými ikonami budov. Přepínání mezi základními obrazovkami Mapa a Skupiny se přesunulo na spodní okraj obrazovky do formy záložek.

V mapě došlo ke změně chování při kliknutí na ikonu budovy. Při kliknutí se pouze zobrazí malé textové pole navázané na ikonu (tzv. informační bublina) se zobrazeným názvem a adresou budovy. Po kliknutí na tuto bublinu se otevře obrazovka s detailem budovy. V rámci tohoto detailu jsou uvedené veškeré informace o budově stejně jako

¹² <https://itunes.apple.com/cz/app/muni-buildings/id1071458549?mt=8>

v aplikaci Budovy MU. Navíc jsou zde dostupná tlačítka „Navigovat“ a „Ukázat na mapě“. Pro navigaci je opět využito výchozího navigačního programu v mobilním zařízení. Tlačítko „Ukázat na mapě“ přesměruje na speciální obrazovku s mapou, kde je nad ikonou zvolené budovy rozbalena informační bublina s názvem a adresou budovy. Nejedná se však o základní obrazovku mapy. Není zde umožněn proklik z bubliny do detailu budovy, ani nejsou zobrazeny záložky pro přechod do základních obrazovek. Z této speciální obrazovky se lze pouze vrátit zpět do detailu dříve zvolené budovy pomocí tlačítka zpět umístěného v horní liště.



Obrázek 9 – Aplikace MUNI Buildings

V seznamu budov bylo změněno chování po kliknutí na budovu, kdy se zobrazí přímo detail budovy.

V nastavení zůstala možnost definování toho, jaké skupiny budov se mají v aplikaci zobrazovat. Přibyla možnost ručního přepínání jazyků mezi češtinou a angličtinou. V aplikaci také byla vyřešena aktualizace dat. V nastavení je pro to vyčleněno tlačítko „Aktualizovat data z webu“. Na dostupnost aktuálnějších dat je uživatel upozorněn červeným vykřičníkem u ikony Nastavení na horní liště v základních obrazovkách Mapa a Skupiny.

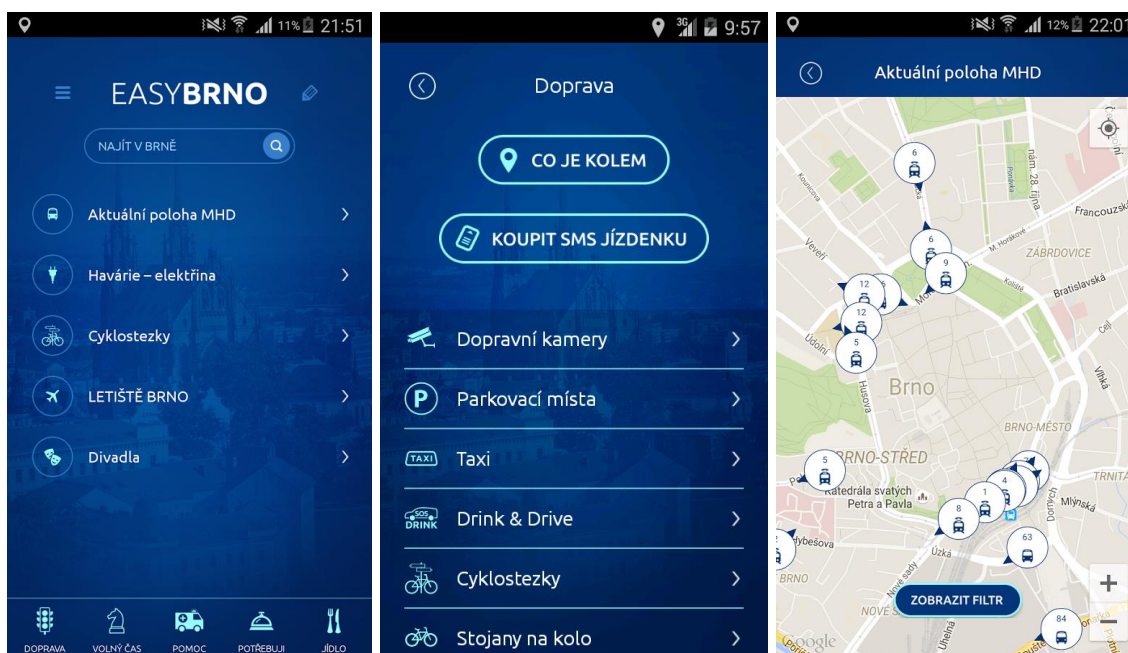
Shrnutí

Aplikace je o čtyři roky novější než její „androidí“ předchůdce. Je použito čistější grafiky a lépe graficky vyřešených ikon budov. Celkově je aplikace díky úpravám chování některých funkcí uživatelsky přívětivější. Důležitým posunem bylo vyřešení aktualizací dat, kdy již není nutné vydávat novou verzi aplikace.

V aplikaci je však velmi nešikovně vyřešeno zobrazení vybrané budovy na mapě, kdy se otevře nová obrazovka s novou mapou bez funkčních prvků, na které je uživatel zvyklý ze základní obrazovky Mapa. Záložky pro přepínání mezi Mapou a Skupinami umístěné na spodní straně obrazovky jsou zobrazeny pouze v těchto dvou základních obrazovkách. V ostatních obrazovkách (nastavení, detail skupiny, detail, budovy) již záložky zobrazeny nejsou, zpět se lze dostat pouze pomocí tlačítka zpět v horní liště. Toto opět vede k jisté dezorientaci uživatele.

4.1.3 Easy Brno

Aplikace Easy Brno¹³ (Obrázek 10) je dalším příkladem aplikace sdružujícím informace o místech v Brně. Aplikace byla vyvinuta v roce 2014–2015 vývojářskou skupinou X Production s.r.o. Easy Brno je dostupné pro operační systémy iOS, Android i Windows Phone. Aplikace získala první místo v kategorii Nejlepší firemní aplikace sponzorované firmou IBM na 15. přehlídce mobilních aplikací AppParade.



Obrázek 10 – Aplikace Easy Brno

Základní obrazovka aplikace obsahuje pole pro fultextové vyhledávání, výpis oblíbených položek a spodní panel se záložkami. Aplikace obsahuje velké množství bodů zájmů, které jsou pomocí záložek rozděleny do pěti kategorií – Doprava, Volný čas, Pomoc, Potřebuji a Jídlo. V každé kategorii jsou pak body zájmů rozděleny do dalších

¹³ <http://www.xproduction.cz/cs/easy-brno-mobilni-aplikace>

skupin. Na mapě se zobrazí buď všechny body celé skupiny (např. všechny bankomaty), nebo jen jediný vybraný bod (např. nejbližší bankomat vybrané banky).

Aplikace je zpracovaná graficky velmi pěkně a čistě. Úvodní stránka s fulltextovým vyhledáváním a možností zobrazení vlastních oblíbených bodů zájmů přispívá k vynikající uživatelské přívětivosti. Aplikace však pokulhává na aktuálnosti, případně úplnosti zobrazovaných dat. Udržení takového množství dat v aktuálním stavu vyžaduje velké úsilí, které této aplikaci pravděpodobně věnováno není. Na webových stránkách vývojáře¹³ se však píše o připravovaném velkém updatu, ve kterém snad dojde i k aktualizaci a doplnění bodů zájmu.

4.2 Požadavky

Požadavky na vytvoření aplikace vzneslo Oddělení facility managementu MU (OFM MU). Požadovaná aplikace má být multiplatformní a má nahradit obě předchozí aplikace vydané na OFM MU – Budovy MU pro Android a MUNI Buildings pro iOS.

Základní požadavky na aplikaci:

- dostupnost pro platformy iOS a Android,
- data uložena v zařízení, aplikace funkční i bez připojení k internetu,
- data aktualizovaná z webu,
- budovy zobrazeny polygonem dle skutečného půdorysu.

Rozšiřující návrhy na aplikaci:

- navigace:
 - pro pěší, auto, MHD,
 - možnost nalézt trasu i z jiného než aktuálního bodu,
- přidat body zájmu (stravovací zařízení, bankomaty, lékař apod.),
- fulltextové vyhledávání v datech o budovách,
- vkládání pozvánek na akce s jejich lokalizací.

4.3 Analýza a návrh aplikace

Aplikaci autor navrhnul s ohledem na zkušenosti získané užíváním starších aplikací OFM MU, které má nová aplikace nahradit. Při návrhu rozložení obrazovek a grafického zpracování se autor také inspiroval aplikací Easy Brno (kapitola 4.1.3) a kladl důraz na jednoduchost a uživatelskou přívětivost aplikace.)

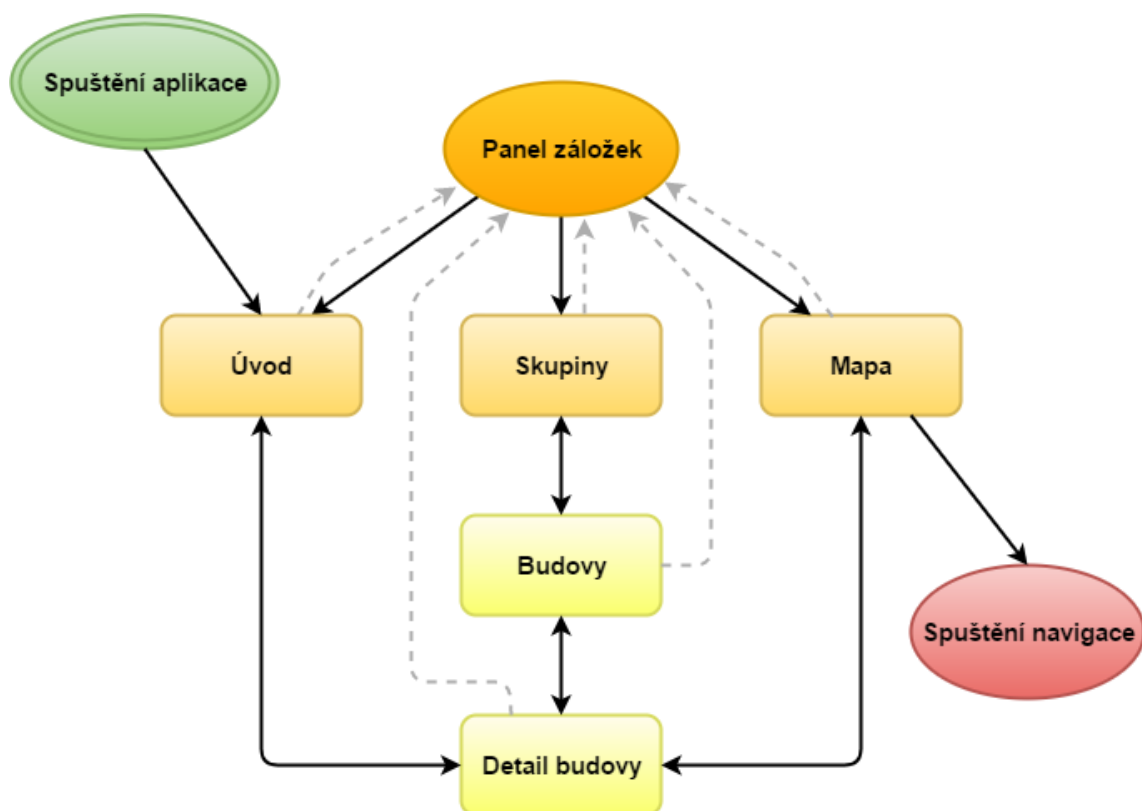
4.3.1 Jazyk aplikace

Aplikace bude dostupná ve dvou jazycích – češtině a angličtině. Volba jazyka bude probíhat automaticky podle jazyka použitého v operačním systému mobilního zařízení.

Při použité češtině bude jazyk aplikace také čeština. Při jakémkoliv jiném jazyce zvoleném v operačním systému bude jazyk aplikace nastaven na angličtinu. Jazyk bude možné v aplikaci ručně přepínat.

4.3.2 Obrazovky

Aplikace bude obsahovat tři základní obrazovky – úvodní obrazovku „Úvod“, obrazovku „Skupiny“ a obrazovku „Mapa“. Hlavním ovládacím prvkem aplikace bude panel záložek umístěný ve spodní části obrazovky, který bude sloužit k přepínání základních obrazovek. Po kliknutí na jednu ze záložek tedy proběhne přesměrování na danou základní obrazovku. Panel záložek bude zobrazený stále – na všech základních obrazovkách i na všech dílčích obrazovkách. Na panelu bude vždy zvýrazněna aktuální aktivní záložka, tedy i v případě dílčích obrazovek patřících pod nějakou záložku. Uživatel bude tedy moci z jakéhokoli stavu aplikace přejít přímo na libovolnou základní obrazovku. V aplikaci budou další dvě dílčí obrazovky patřící do záložky „Skupiny“. Bude to obrazovka „Budovy“ a obrazovka „Detail budovy“. Logika přechodů mezi obrazovkami je znázorněna na Obrázek 11.



Obrázek 11 – Logika přechodů mezi obrazovkami

Základní obrazovka „Úvod“

Úvodní obrazovka bude plnit funkci uvítací obrazovky. Bude použito grafického pozadí a v horní části obrazovky bude zobrazený název aplikace. Ve spodní části obrazovky bude jako u všech ostatních obrazovek umístěn panel záložek. Střední část obrazovky bude určena pro vlastní uživatelské přizpůsobení aplikace v podobě zobrazení oblíbených

budov uživatele. Po kliknutí na zobrazenou oblíbenou budovu proběhne přesměrování na obrazovku „Detail budovy“. V případě implementace fulltextového vyhledávání bude vyhledávací pole umístěno pod zobrazeným názvem aplikace. Při kliknutí do tohoto pole se zobrazí klávesnice. Zároveň zmizí výpis oblíbených budov, místo kterého se budou vypisovat výsledky hledání.

Na úvodní obrazovce bude v horní části také ikona pro rozbalení bočního menu vyjíždějícího z levé strany. V tomto menu bude základní nastavení aplikace jako např. ruční přepínání jazyků. Při dalším rozšiřování aplikace zde bude prostor pro odkazy na další obrazovky aplikace – např. pokročilejší nastavení, stránka s popisem aplikace apod.

V rámci obrazovky „Úvod“ bude při každém jejím otevření probíhat logika zajišťující aktualizaci lokální databáze. Ta při aktivním internetovém připojení odešle dotaz na server zjišťující dostupnost aktuálnější verze databáze. Při kladné odpovědi se na úvodní obrazovce pod názvem aplikace zobrazí tlačítko umožňující stažení nové verze databáze z webového serveru. Podrobněji popsáno v kapitole 4.4.

Základní obrazovka „Skupiny“

Tato obrazovka již nebude mít grafické pozadí. V horní části bude zobrazen stavový řádek s názvem aktuální obrazovky.

Celý zbytek obrazovky bude obsahovat pouze výpis skupin budov. Rozdělení do skupin bude vycházet z organizačních jednotek univerzity (typicky fakulty, rektorát, knihovny, ...). U každé skupiny bude zobrazena ikona a název. Výpis bude seřazen podle názvu skupin v příslušném jazyce. Při kliknutí na skupinu proběhne přesměrování na stránku „Budovy“.

Obrazovka „Budovy“

Obrazovka „Budovy“ bude velice podobná obrazovce „Skupiny“. Bude patřit pod záložku „Skupiny“, tzn. V panelu záložek bude zvýrazněná záložka „Skupiny“. Po kliknutí na tuto záložku proběhne běžné přesměrování na základní obrazovku „Skupiny“.

Ve stavovém řádku přibude na levé straně tlačítko zpět, provádějící přesměrování zpět na předchozí obrazovku. V tomto případě tedy vždy proběhne přesměrování na obrazovku „Skupiny“. Místo názvu obrazovky bude ve stavovém řádku zobrazen název vybrané skupiny budov.

Zbytek obrazovky bude obsahovat výpis budov dané skupiny. Každý záznam se bude skládat z názvu budovy a její adresy. Výpis bude seřazen podle adres. Po kliknutí na budovu proběhne přesměrování na obrazovku „Detail budovy“.

Obrazovka „Detail budovy“

Tato obrazovka bude graficky podobná předchozím dvěma obrazovkám. Bude obsahovat informace o dané budově.

Ve stavovém řádku bude opět zobrazeno tlačítko zpět a dále název dané budovy. Do „Detailu budovy“ se lze dostat ze tří různých obrazovek – typicky z obrazovky „Budovy“, poté přes oblíbené budovy z obrazovky „Úvod“ a dále přes ikonu budovy

v mapě z obrazovky „Mapa“. Tlačítko zpět ve stavovém řádku provede vždy přesměrování na předchozí obrazovku, tzn. na tu obrazovku, ze které uživatel přistoupil do „Detailu budovy“.

V hlavní části obrazovky bude zobrazena ikona skupiny, do které budova patří, společně s názvem budovy. Dále pak adresa budovy, ovládací tlačítka umožňující provádět akce s danou budovou a popis budovy. Ovládací tlačítka budou dvě. První bude sloužit k přidání či odebrání budovy ze seznamu oblíbených budov. Informace o tom, zda budova je zařazená v oblíbených či nikoliv, bude uživateli zprostředkována pomocí popisu tlačítka (Přidat do oblíbených / Odebrat z oblíbených) a také pomocí výrazné grafické změny podoby tlačítka korespondující s popisem tlačítka. Po kliknutí na tlačítko „Přidat do oblíbených“ se identifikační číslo budovy uloží do seznamu oblíbených budov v lokální databázi. Zároveň se tlačítko změní na „Odebrat z oblíbených“. Analogicky po kliknutí na tlačítko „Odebrat z oblíbených“. Druhé tlačítko s názvem „Zobrazit na mapě“ bude sloužit k přesměrování do základní obrazovky „Mapa“, na které se provede přiblížení na danou budovu a označení ikony této budovy. Popis budovy bude realizován pomocí naformátovaného HTML textu.

Základní obrazovka „Mapa“

Maximální možná plocha této obrazovky bude využita pro zobrazení mapy. Obrazovka tedy nebude obsahovat stavový řádek. V mapě budou zobrazeny statické půdorysy budov a dynamické ikony budov ve formě špendlíků do mapy. Po kliknutí na ikonu se u ikony zobrazí informační bublina obsahující název budovy a text informující uživatele, že po kliknutí na tuto informační bublinu získá více informací o zvolené budově. Po kliknutí na informační bublinu proběhne přesměrování do obrazovky „Detail budovy“.

Do obrazovky „Mapa“ bude možné přejít buď pomocí záložky v panelu záložek nebo z obrazovky „Detail budovy“. Při přechodu pomocí panelu záložek bude mapa zobrazena ve stejném stavu, v jakém se nacházela při posledním opuštění obrazovky. Při prvotním zobrazení mapy bude mapa oddálená na celé město Brno. Při přechodu z obrazovky „Detail budovy“ se mapa vždy přiblíží na danou budovu a ikona této budovy bude zvýrazněna zobrazením informační bubliny.

Mapa bude nabízet základní možnosti ovládání, na které jsou uživatelé zvyklí z běžných mobilních mapových služeb:

- ovládání pomocí gest jedním prstem:
 - posun mapy,
- ovládání pomocí gest dvěma prsty:
 - přiblížení / oddálení,
 - rotace,
 - případně i zkosení,
- tlačítka pro přiblížení a oddálení mapy,
- tlačítko pro zobrazení aktuální polohy mobilního zařízení.

4.4 Aktualizace dat aplikace

Aktualizace dat bude v navrhované mobilní aplikaci probíhat při aktivním internetovém připojení pomocí komunikace s webovým serverem. Data budou uložena ve webové databázi a odtud budou pomocí webové serverové služby stahována do lokálních databází navrhované aplikace v mobilních zařízeních.

4.4.1 Mobilní aplikace

Jak již bylo uvedeno v kapitole 4.3.2, logika zajišťující aktualizaci dat v lokální databázi bude probíhat při každém otevření obrazovky „Úvod“.

Při otevření obrazovky „Úvod“ bude vždy odeslán dotaz na serverovou službu „metadata“ zjišťující aktuální verzi hlavní databáze. Při neaktivním internetovém připojení se nepodaří dotaz odeslat, tedy aplikace nepřijme žádnou odpověď. V takovém případě proces aktualizace dat končí. Při aktivním internetovém připojení vrátí serverová služba číslo aktuální verze hlavní databáze. Pokud bude toto číslo verze větší než číslo verze lokální databáze, zobrazí se na úvodní obrazovce tlačítko „Stáhnout aktuální verzi“ umožňující aktualizaci lokální databáze. Pokud se během odesílání dotazu na serverovou službu vyskytne jakákoliv chyba, aplikace proces aktualizace neprovede a uživatele na to upozorní pomocí vyskakovacího okna.

Po kliknutí na tlačítko „Stáhnout aktuální verzi“ se odešle dotaz na serverovou službu „data“. Při neaktivním internetovém připojení bude na tuto skutečnost uživatel upozorněn pomocí vyskakovacího okna přímo v aplikaci. Při aktivním internetovém připojení serverová služba vrátí všechna data z hlavní databáze. V lokální databázi se nahradí původní data novými daty přijatými ze serverové služby. Poté se v lokální databázi uloží číslo nové verze databáze. Pokud během komunikace se serverem dojde k chybě, uživatel na to bude opět upozorněn pomocí vyskakovacího okna v aplikaci.

4.4.2 Databáze

Hlavní databáze

Hlavní databázi pro ukládání dat celého řešení bude databáze uložená na databázovém serveru, která bude trvale dostupná prostřednictvím serverových služeb. Bude obsahovat tabulky „metadata“, „buildings“, „buildings_2d“ a „groups“.

Tabulka „metadata“ bude obsahovat informace o aktualizacích databáze. Každý záznam bude odpovídat jedné dokončené aktualizaci databáze a bude obsahovat identifikační číslo (ID) aktualizace, datum a stručný popis aktualizace.

Tabulka „buildings“ bude obsahovat záznamy o všech budovách zobrazených v mobilní aplikaci. Každá zobrazená ikona v mapě mobilní aplikace bude odpovídat jednomu záznamu v tabulce „buildings“. To znamená, že pro jednu budovu může být v tabulce více záznamů. Příkladem může být budova, která by patřila do organizační skupiny budov nějaké fakulty a zároveň se by se v téže budově nacházela knihovna patřící do organizační skupiny knihoven. Každý záznam v tabulce bude obsahovat jedinečné identifikační číslo záznamu, název a popis budovy v českém a anglickém jazyce, poštovní

adresu, webovou adresu URL, příslušnost ke skupině budov, označení budovy kódem jedinečným pro budovu (nikoliv jedinečným pro daný špendlík) a souřadnice špendlíku na mapě.

Tabulka „buildings_2d“ bude obsahovat půdorysy budov. Každý záznam v tabulce bude odpovídat jedné budově a bude obsahovat pole souřadnicových bodů definujících půdorys budovy a označení budovy jedinečným kódem.

Tabulka „groups“ bude obsahovat informace o organizačních skupinách, do kterých budou přiřazeny jednotlivé budovy. Každý záznam v tabulce bude odpovídat jedné skupině budov a bude obsahovat jedinečné identifikační číslo skupiny, název skupiny v českém a anglickém jazyce, URL adresu organizační skupiny v rámci univerzity a označení skupiny krátkým jedinečným textovým identifikátorem (např. „lib“ pro knihovny). Pomocí tohoto identifikátoru se budou přiřazovat jednotlivé budovy do příslušných skupin budov.

Lokální databáze

Lokální databáze bude uložena v každém mobilním zařízení, na kterém bude nainstalována navržená aplikace. Tato databáze bude vytvořena již při instalaci aplikace. Bude obsahovat všechna data zobrazovaná v aplikaci. Aplikace tedy poběží i v offline režimu bez aktivního internetového připojení.

Struktura lokální databáze bude obdobná struktuře hlavní databáze. Bude obsahovat tabulky „config“, „buildings“ a „groups“.

Tabulka „groups“ bude přesnou kopií stejnojmenné tabulky z hlavní databáze.

Tabulka „buildings“ bude také kopií stejnojmenné tabulky z hlavní databáze. Navíc však bude obsahovat půdorysy budov z tabulky „buildings_2d“ z hlavní databáze (viz následující kapitola 4.4.3).

Tabulka „config“ bude obsahovat informace o aktuální konfiguraci aplikace. Bude zde uloženo číslo aktuální verze lokální databáze a pole oblíbených budov uživatele. V této tabulce bude také prostor pro další případné konfigurační hodnoty, které se mohou objevit při budoucích rozšířeních celé aplikace. Například to můžou být různé hodnoty uživatelského nastavení zobrazení mapy a podobně.

4.4.3 Webový server

Webový server bude zajišťovat spojení mezi hlavní databází a samotnou instancí navrhované mobilní aplikace. Bude obsahovat dvě serverové služby – službu „metadata“ a službu „data“.

Služba „metadata“ bude sloužit pro zjištění čísla aktuální verze dat v hlavní databázi. Bude vracet poslední záznam vložený do tabulky „metadata“, tzn. záznam s nevyšším ID.

Služba „data“ bude sloužit ke stažení aktuálních dat z hlavní databáze. Bude vracet kompletní obsah tabulek „groups“ a „buildings“. Záznamy v tabulce „buildings“ budou navíc rozšířeny o půdorysy budov z tabulky „buildings_2d“, které budou propojeny na základě shodného kódu budovy.

5 IMPLEMENTACE APLIKACE

5.1 Výběr vhodného vývojového prostředí

Pro implementaci navržené aplikace s ohledem na vlastní zkušenosti s webovými technologiemi autor zvolil vývojové prostředí Ionic. Během implementace navržené aplikace nebyla dostupná finální verze Ionic 2.0. Aplikace je tedy implementována v prostředí Ionic verze 1.0.

5.1.1 Instalace SDK mobilních platforem

Pro vytváření aplikací v Ionicu je potřeba nainstalovat SDK platform, pro které bude aplikace vytvářena.

Android SDK pro platformu Android lze nainstalovat v operačním systému Windows, Mac OS X i Linux. Instalační soubory pro Android SDK jsou dostupné ke stažení na oficiálních stránkách¹⁴. Pro vývoj Android aplikací v Ionicu je Android Studio zcela zbytečné. Lze tedy využít i odlehčenou verzi SDK – Android stand-alone SDK. Po instalaci SDK je nutné přidat cestu k Android nástrojům do systémové proměnné PATH. Dalším krokem je otevření nástroje Android Manager, který nabízí instalaci různých balíčků pro Android SDK. Pro vývoj v Ionicu je potřeba těchto balíčků:

- Android SDK Tools,
- Android SDK Platform-tools,
- Android SDK Build-tools,
- Android Support Library.

Dále je nutné nainstalovat SDK Platform pro libovolnou verzi OS Android.

iOS SDK je možné nainstalovat pouze v operačním systému Mac OS X. Instalace se provede jednoduše přes AppStore nainstalováním vývojového prostředí Xcode. Zbytek postupu je shodný jako u Android SDK.

Aplikace byla vyvíjena v operačním systému Windows 10, kde tedy bylo možné vytvořit finální aplikační balíček pouze pro platformu Android. Pro vytvoření balíčku pro platformu iOS je nutné využít osobní počítač s operačním systémem, na kterém bude nainstalovaný iOS SDK a vývojové prostředí Ionic. Překlad a vydání aplikací pro obě platformy je dále popsán v kapitole 5.5.

5.1.2 Instalace Ionicu

Instalace Ionicu vyžaduje nejprve instalaci softwarového systému Node.js. Instalační balík lze stáhnout z oficiálních stránek Node.js¹⁵ a nainstalovat podle instrukcí uvedených

¹⁴ <https://developer.android.com/studio/index.html>

¹⁵ <https://nodejs.org/en/>

tamtéž. Společně s Node.js se nainstaluje i balíčkovací systém npm, díky kterému lze poté snadno globálně nainstalovat Ionic příkazem:

```
> npm install -g ionic
```

5.1.3 Práce s Ionicem

Vývoj v Ionicu je řízen z terminálu pomocí rozhraní CLI. V následujícím textu budou přestaveny základní příkazy tohoto rozhraní.

Vygenerování nové aplikace se provede příkazem:

```
> ionic start nazev_aplikace blank
```

Tímto se vytvoří Ionic projekt podle šablony blank (prázdný projekt). Dále lze využít šablonu tabs (záložky) a šablonu sidemenu (postranní menu).

Takto vytvořená aplikace je již plně funkční a obsahuje demonstrativní ukázky některých UI komponent.

V CLI lze také vytvořit Ionic projekt přímo z aplikace vytvořené ve webovém nástroji Ionic Creator. To se provede pomocí jedinečného ID aplikace z Ionic Creator pomocí příkazu:

```
> ionic start nazev_aplikace creator:id_aplikace
```

Pro spuštění aplikace v prohlížeči slouží příkaz:

```
> ionic serve [-l]
```

Volitelným parametrem „-l“ se aplikace spustí v prohlížeči v nástroji Ionic Lab, který zobrazí spuštěnou aplikaci dvakrát – pro každou platformu zvlášť.

Pro spuštění aplikace v emulátoru pak slouží příkaz:

```
> ionic emulate [android/ios]
```

Emulátory jsou různé pro obě dostupné platformy, proto je nutné v tomto příkazu specifikovat platformu, pro kterou má být emulátor spuštěn.

Aplikace lze spouštět také v tzv. ladícím režimu přímo na mobilních zařízeních. To se provede příkazem:

```
> ionic run [android/ios]
```

Po vytvoření projektu je třeba do něj přidat všechny soubory a nástroje potřebné pro sestavování výsledného aplikačního balíčku dané platformy. Pro toto slouží příkaz:

```
> ionic platform add [android/ios]
```

Pro přidání pluginu do projektu Ionicu pak slouží podobný příkaz:

```
> ionic plugin add nazev_pluginu
```

Pro vyvolání nápovědy Ionicu slouží příkaz:

```
> ionic help
```

Rozhraní CLI obsahuje mnoho dalších příkazů. Jejich přehled je dostupný v dokumentaci Ionicu [18].

5.2 Implementace navržené aplikace

Implementace aplikace vychází z analýzy a návrhu aplikace z kapitoly 4.

5.2.1 Struktura projektu

Zdrojové soubory aplikace jsou uloženy v adresáři aplikace v podadresáři *www*. Zde jsou přehledně rozděleny podle typů souborů do podadresářů *css*, *img*, *js*, *lib* a *templates*. V adresáři *www* se nachází také základní soubor aplikace *index.html*, který odkazuje na další zdrojové soubory CSS a JavaScript.

V adresáři *css* se nacházejí soubory CSS s kaskádovými styly použitými v aplikaci.

Adresář *img* obsahuje obrázky použité v aplikaci – pozadí úvodní obrazovky a ikony pro jednotlivé skupiny budov pojmenované pomocí textového identifikátoru dané skupiny. Takto je pak ikona přiřazena k dané skupině.

Adresář *js* obsahuje všechny javascriptové soubory implementované autorem aplikace:

- zdrojové kódy pro zajištění běhu aplikace – *app.js*,
- zdrojové kódy pro definování stavů aplikace – *routes.js*,
- kontrolery obsahující řídicí logiku jednotlivých obrazovek aplikace – podadresář *controllers*,
- služba *LokiService.js* zajišťující funkci lokální databáze,
- definice základních proměnných aplikace (např. URL adresa webového serveru) – *default_config.js*,
- soubor s překlady použitých textových řetězců – *translations.js*.

V adresáři *lib* jsou naimportované všechny knihovny použité v aplikaci. O obsah tohoto adresáře se stará rozhraní CLI.

V adresáři *templates* jsou pak HTML soubory pro všechny obrazovky aplikace. Každá obrazovka se skládá z grafického popisu pomocí HTML a CSS a z popisu logiky obrazovky pomocí javascriptového Angular kontroleru a Angular konstrukcí vložených přímo do HTML souboru. Ukázky zdrojového kódu jsou v Příloze 1.

5.2.2 Mapová API

Pro zobrazení mapy v aplikaci je využito pluginu Cordova GoogleMaps plugin¹⁶, který je dostupný pro platformy iOS a Android. Plugin využívá API Google Maps – Google Maps Android API a Google Maps SDK for iOS. Pomocí tohoto pluginu lze vytvořit nativní Google mapu s využitím JavaScriptu.

Veškerá práce s mapou v implementované aplikaci je zapsána ve zdrojovém souboru *mapaCtrl.js* v sekci kontrolerů.

Navigace je v aplikaci řešena pomocí funkce vložené Google mapy. Po kliknutí na ikonu budovy se ve spodní části mapy zobrazí tlačítko umožňující přesměrování do aplikace Google Maps s přednastaveným cílovým bodem na souřadnicích dané budovy.

Pomocí aplikace Google Maps lze také do mobilního zařízení stáhnout offline verzi mapy vybraného území. Tato stažená mapa je poté využita i v aplikacích pracujících s Google Maps API. Mimo tuto možnost dokáží mapy zobrazené pomocí Google Maps API automaticky uchovávat mapové podklady dříve zobrazených území, které jsou potom alespoň částečně zobrazeny i v případě neaktivního internetového připojení.

5.2.3 Lokální databáze

Pro ukládání dat v mobilním zařízení je využito dokumentově orientované javascriptové databáze LokiJS¹⁷. LokiJS je dostupná pro mobilní prostředí Cordova a tedy i pro Ionic. LokiJS nahrazuje v Cordově databáze SQLite.

Databáze LokiJS podporuje indexování, dotazování a filtrování datových kolekcí. Datová kolekce je obměnou databázové SQL tabulky. Data jsou uložena jako objekty JavaScriptu a pro trvalé uložení na disku jsou tyto objekty serializované na JSON. Pro dotazování se nevyužívají SQL dotazy, nýbrž rychlejší nástroje založené na automatickém indexování dat a využití binárních vyhledávacích algoritmů.

V implementované aplikaci je práce s lokální databází LokiJS implementována ve službě *LokiService.js*. V této službě je řešeno vytvoření lokální databáze, její inicializace, otevření celé databáze a otevření zvolené datové kolekce. Službu je možné importovat a využít v kterémkoliv zdrojovém javascriptovém souboru. Samotná práce s daty jako přidávání, mazání a aktualizace záznamů je řešena přímo v příslušných javascriptových kontrolerech.

5.2.4 Aktualizace dat

Návrh procesu aktualizace dat je podrobně popsán v kapitole 4.4. Přesně podle tohoto návrhu byla provedena implementace. Pro odesílání dotazů na serverové služby je využito asynchronní AngularJS služby *\$http*. Data vrácená ze serverových služeb jsou uložena do javascriptových objektů a poté dále zpracována, případně uložena do lokální databáze.

¹⁶ <https://github.com/mapsplugin/cordova-plugin-googlemaps-doc>

¹⁷ <http://lokijis.org/>

5.3 Implementace serverových služeb

Obě navržené serverové služby „data“ i „metadata“ jsou implementované v programovacím jazyce PHP a jsou volně dostupné na webovém serveru. Pro připojení k hlavní databázi je využito PHP knihovny MySQLi. Ukázka zdrojového kódu serverové služby je v Příloze 2.

Obě serverové služby vracejí text strukturovaný do datové podoby formátu JSON. Jako první objekt je v obou případech uveden stav „state“ nabývající hodnot 0 a 1. Tento stav určuje, zda vše v serverové službě proběhlo v pořádku. Například pokud se serverové službě nepodaří připojit do hlavní databáze, serverová služba vrátí stav 0. Pokud vše proběhne v pořádku, bude stav nastaven na hodnotu 1.

Dalšími objekty jsou pak samotná data. Ve službě „metadata“ jsou tato data označena identifikátorem „data“ a obsahují záznam s nejvyšším ID z tabulky „metadata“ hlavní databáze. Celý výstup serverové služby „metadata“ může vypadat například takto:

```
{
  "state":1,
  "data":
    {
      "id":"39",
      "datum":"2017-05-23 09:33:08",
      "popis":"odstraneni testovacich budov"
    }
}
```

V serverové službě „data“ jsou výstupní data rozdělená do dvou objektů – „groups“ a „buildings“. Objekt „groups“ obsahuje všechny záznamy z tabulky „groups“ hlavní databáze. Objekt „buildings“ pak obsahuje všechny záznamy z tabulky „buildings“ hlavní databáze. Pokud k danému záznamu existuje půdorys v tabulce „buildings_2d“ v hlavní databázi, je tento půdorys přidán k záznamu v objektu „buildings“. Výstup serverové služby „data“ může vypadat například takto:

```
{
  "state":1,
  "groups":
    [
      {údaje_první_skupiny},
      {údaje_druhé_skupiny},
      ...
    ],
  "buildings":
    [
      {údaje_první_budovy},
      {údaje_druhé_budovy},
      ...
    ]
}
```

5.4 Popis hlavní databáze

Hlavní databáze je typu MySQL. Tabulky v databázi odpovídají návrhu z kapitoly 4.4.2. Databázi lze editovat přímo pomocí webového klienta phpMyAdmin. Vhodnějším způsobem zvláště pro rozsáhlejší editaci databáze je její stažení pomocí webového klienta, následná editace ve vhodném pokročilejším editačním databázovém programu a závěrečné nahrání upravené databáze na databázový server opět pomocí klienta phpMyAdmin.

5.5 Publikování aplikace

Pro publikování aplikace je nejprve nutné vygenerovat výsledný aplikační balíček. To se v Ionicu provede pomocí příkazu:

```
> ionic build [android/ios] --release
```

Výsledný aplikační balíček bude sestavený podle nastavení uvedených v souboru *config.xml* – název aplikace, popis, číslo verze apod. Soubor *config.xml* je uložený v hlavní složce projektu aplikace.

Dále bude potřeba aplikační balíček tzv. podepsat pomocí podpisového klíče a optimalizovat jej pro potřeby publikování v aplikačních obchodech.

V případě platformy Android se podepsání a optimalizace aplikačního balíčku u Ionic aplikací provádí v terminálu. Pomocí nástroje keytool, který je součástí JDK (Java Development Kit), lze vytvořit nový podpisový klíč. K podepsání aplikačního balíčku se poté použije nástroj jarsigner, který je také součástí JDK. Pro vytvoření finálního aplikačního balíčku určeného pro publikování je jej nutné ještě optimalizovat pomocí nástroje zipalign, který je součástí Android SDK. Takto vytvořený aplikační balíček lze poté publikovat v obchodě Google Play pomocí webové aplikace Google Play Console¹⁸.

V případě platformy iOS se podepsání aplikace podpisovým klíčem, vytváření podpisových klíčů i optimalizace aplikačních balíčků provádí ve vývojovém prostředí Xcode. Pro tento účel je nutné prostředí Xcode propojit s vývojářským účtem Apple pomocí Apple ID. Finální aplikační balíček lze poté publikovat do obchodu AppStore pomocí webové aplikace iTunes Connect¹⁹.

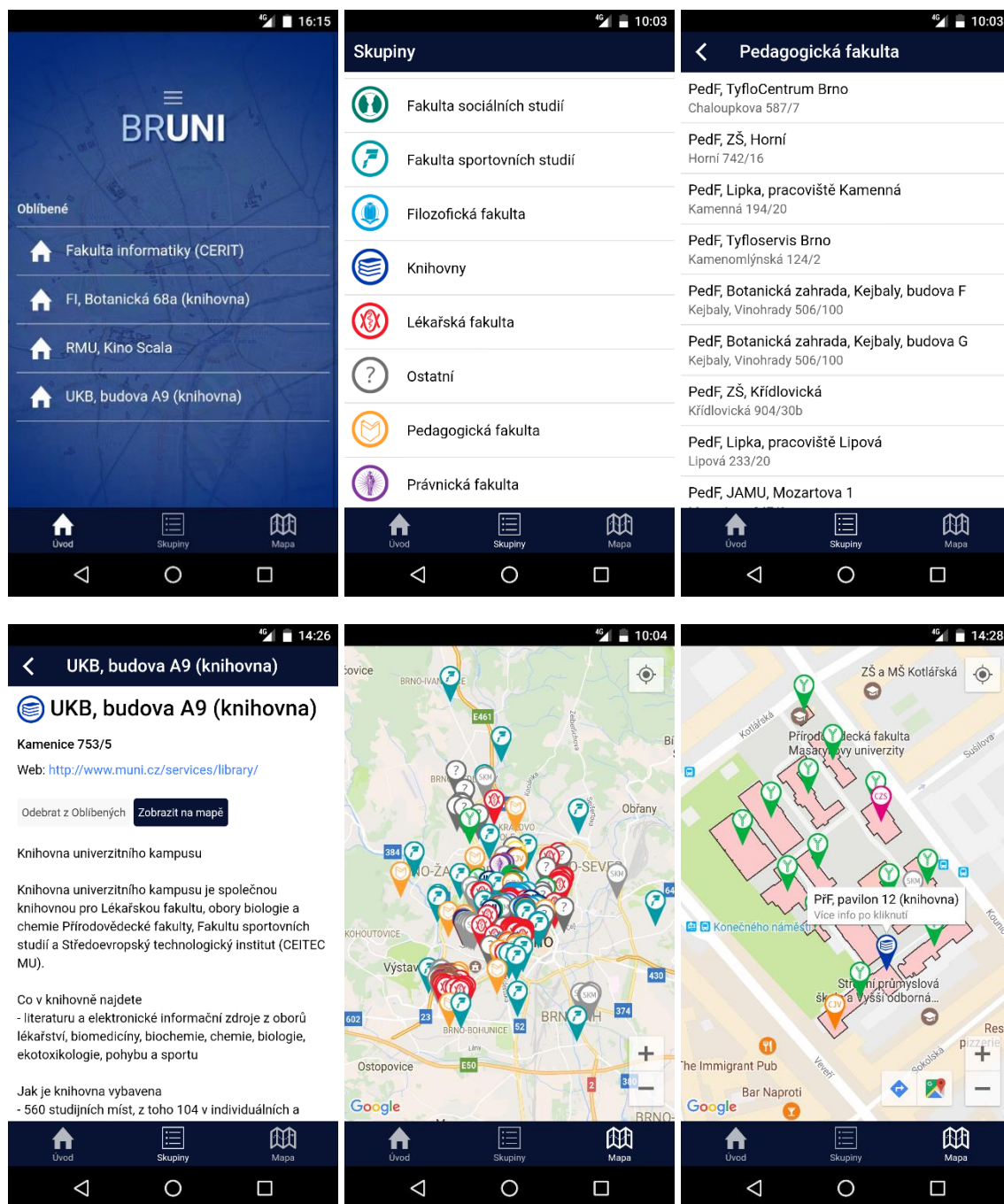
V případě aktualizace aplikace je nutné v souboru *config.xml* změnit číslo verze a aplikaci sestavit znovu, včetně podepsání podpisovým klíčem a optimalizace aplikačního balíčku. [18]

¹⁸ <https://play.google.com/apps/publish/>

¹⁹ <https://itunesconnect.apple.com/>

5.6 Výsledná aplikace

Výsledná aplikace je zobrazena na Obrázek 12 a podrobněji také v Příloze 3. Aplikace dostala pracovní název „BRUNI – brněnské univerzity“. Chování aplikace odpovídá návrhu z kapitoly 4. Aplikace je připravená pro nasazení na jednu univerzitu. Nyní jsou v aplikaci použita data budov Masarykovy Univerzity.



Obrázek 12 – Implementovaná aplikace BRUNI

5.7 Testování

Pro účely testování aplikace na mobilních zařízeních s platformou Android byla vydána beta verze aplikace přímo v obchodě Google Play. Aplikace není v obchodě veřejně dostupná. Přístup k ní má pouze uzavřený okruh testerů. Těm je umožněno si aplikaci stáhnout běžným způsobem z obchodu Google Play do svého mobilního zařízení.

Pro účely testování na mobilních zařízeních s platformou iOS byla aplikace distribuována pomocí speciální služby Ionic View, pomocí které lze sdílet aplikace mezi testery bez nutnosti vydání aplikačního balíčku.

5.7.1 Android

Výpis testů seřazený podle verze operačního systému Android:

- Android 2.3.7 – modifikace CyanogenMod 7:
 - Testovaná zařízení: ZTE Blade,
 - Výsledek testu: aplikaci nelze nainstalovat.
- Android 4.0:
 - Testovaná zařízení: Samsung Galaxy S2 duos,
 - Výsledek testu: aplikaci nelze nainstalovat.
- Android 4.1.2:
 - Testovaná zařízení: Samsung Galaxy Ace 2,
 - Výsledek testu: aplikace po spuštění zobrazí pouze bílou obrazovku.
- Android 4.4.2 – modifikace:
 - Testovaná zařízení: Huawei Ascend P6,
 - Výsledek testu: aplikace po spuštění zobrazí pouze bílou obrazovku.
- Android 6.0:
 - Testovaná zařízení: Huawei P8 Lite,
 - Výsledek testu: správná funkčnost.
- Android 5.1:
 - Testovaná zařízení: Lenovo Tab 2 A8-50F (tablet),
 - Výsledek testu: správná funkčnost.
- Android 6.0.1:
 - Testovaná zařízení: Samsung Galaxy J5, Google Nexus 5,
 - Výsledek testu: správná funkčnost.
- Android 6.0.1 – modifikace CyanogenMod 13:
 - Testovaná zařízení: Samsung Galaxy S5 mini,
 - Výsledek testu: správná funkčnost.

Minimální verze operačního systému Android je pro Ionic aplikace Android 4.1. Starší verze systému Android nejsou podporovány. Proto v prvních dvou testech obchod Google Play nepovolil instalaci.

Na Androidech 4.1.x až 4.4.x se po spuštění aplikace Bruni zobrazí pouze bílá obrazovka. Jiné testovací aplikace implementované autorem v Ionicu vykazují správnou

funkčnost. Bílá obrazovka je pravděpodobně způsobená využitím některých funkcí javascriptové knihovny Angular 2 nebo možná použitím některých pluginů, které nejsou podporovány ve WebView dostupném na zařízeních se systémem Android verze 4.1.x až 4.4.x. Tento problém by se měl vyřešit přidáním vlastního WebView přímo do aplikace – Crosswalk²⁰, případně definováním minimální možné verze systému Android na verzi 5.0. Tím by ale aplikace přišla o potenciální uživatele.

5.7.2 iOS

V systému iOS bylo testování možné pouze přes testovací prostředí Ionic View. Test byl proveden pouze na jednom mobilním zařízení se systémem iOS:

- iOS 10.3.1 (testováno přes testovací prostředí Ionic View):
 - Testovaná zařízení: Apple iPad Air 2,
 - Výsledek testu: aplikace funkční, chyba zobrazení ikon budov v mapě.

V testovacím prostředí Ionic View aplikace funguje, avšak ne vše zcela správně. Samotná aplikace a všechny ovládací prvky fungují dle očekávání. Zobrazení textových polí i obrázků taktéž. Práce s mapou také probíhá v pořádku. V mapě se správně zobrazují polygonové půdorysy budov. Ovšem již se zde nezobrazují obrázkové ikony budov (špendlíky). Při použití výchozích ikon Google Maps API se tyto zobrazí správně. Tato chyba je tedy pravděpodobně způsobena jinou adresářovou strukturou nebo jiným omezením v testovacím prostředí Ionic View v kombinaci s využitím Google Maps API. Aplikace tedy není schopná nalézt zdrojové obrázky ikon a proto se na mapě žádné nezobrazí. Při testování bylo vyzkoušeno několik variant cest ke zdrojovým obrázkům, avšak bez úspěchu. Správné cesty bude nutné ověřit po plnohodnotném vydání aplikace mimo testovací prostředí Ionic View.

²⁰ <http://www.arjuns.com/html5/improve-speed-ionic-app-using-crosswalk-part-3/>

6 ZHODNOCENÍ A DISKUZE

Při srovnání implementované aplikace BRUNI s aplikacemi dříve vydanými na OFM MU dochází v aplikaci BRUNI ke značnému posunu v oblasti uživatelské přívětivosti. Největším přínosem je zejména přehledná úvodní obrazovka s možností zobrazení oblíbených budov, ale také jednoduché ovládání díky stále zobrazeným záložkám. Velkým přínosem je také sjednocení dvou předchozích aplikací do jedné multiplatformní aplikace. V této práci bylo otestováno zejména vydání aplikace pro platformu Android. Pro platformu iOS bude nutné aplikaci vydat v operačním systému Mac OS X a následně aplikaci řádně otestovat.

Zveřejnění aplikace v aplikačních obchodech je v rámci Masarykovy Univerzity plánováno před začátkem akademického roku 2017/2018. Do této doby plánuje autor práce dokončit vydání aplikace i pro platformu iOS. Autor také plánuje aplikaci před jejím zveřejněním rozšířit o další funkce.

Zejména se jedná o problém navigace k vybrané budově. V aktuální verzi aplikace je toto řešeno spuštěním navigační aplikace Google Maps. Chování při nedostupnosti aplikace Google Maps autor s ohledem na plánované rozšíření netestoval. Připravovaným rozšířením je doplnění tlačítka „Navigovat“ na obrazovku „Detail budovy“. Po kliknutí na toto tlačítko by se otevřela výchozí navigační aplikace v mobilním zařízení, případně by se přímo v rámci aplikace BRUNI otevřela nabídka s možností výběru mezi dostupnými navigačními aplikacemi v mobilním zařízení. Varianta s nabídkou výběru navigačních aplikací již byla autorem otestována v rámci speciální testovací aplikace, avšak do aplikace BRUNI zatím tato možnost doplněna nebyla.

Dalším plánovaným rozšířením je přidání fulltextového vyhledávání v datech o budovách a skupinách budov. Návrh fulltextového vyhledávání byl popsán v kapitole 4.3.2 v sekci Základní obrazovka „Úvod“. Fulltextové vyhledávání již také bylo autorem otestováno ve speciální testovací Ionic aplikaci.

Posledním rozšířením, které je plánováno implementovat ještě před zveřejněním aplikace, je přidání nové obrazovky „Nastavení“. Odkaz na tuto obrazovku bude z bočního menu. Obrazovka „Nastavení“ bude zpřístupňovat uživateli další možná nastavení aplikace. Zejména se bude jednat o určení skupin budov, které se budou v aplikaci zobrazovat, a které nikoliv. Obrazovka je již v aplikaci připravená, avšak funkčnost nastavení zatím implementována nebyla. Obrazovka je tedy v aktuální verzi aplikace skrytá.

Další rozšíření uvedená v sekci Rozšiřující návrhy na aplikaci v kapitole 4.2 jsou plánovaná až jako případné aktualizace aplikace po jejím zveřejnění v aplikačních obchodech. Tato rozšíření budou implementována s ohledem na aktuální požadavky uživatelů na aplikaci.

Autor by rád aplikaci také rozšířil o budovy ostatních brněnských univerzit. Pro toto rozšíření by však bylo nutné aplikaci náležitě upravit návrhem a implementací rozcestníku univerzit.

7 ZÁVĚR

V úvodní části této práce byly popsány možnosti vývoje multiplatformních aplikací. Možnosti byly rozděleny na tři přístupy k vývoji – vývoj nativních aplikací, vývoj hybridních aplikací a vývoj mobilních webů. Dále byly uvedeny a popsány příklady dostupných vývojových prostředí pro vývoj nativních a hybridních aplikací.

V druhé části práce byl popsán návrh a implementace mobilní aplikace BRUNI usnadňující orientaci v prostředí velké univerzity. Návrh aplikace byl proveden s ohledem na zkušenosti získané při používání dříve vydaných aplikací na Masarykově Univerzitě pro jednotlivé platformy Android a iOS. Vytvořená aplikace BRUNI by měla tyto aplikace nahradit.

Pro implementaci aplikace byl zvolen přístup vývoje hybridních aplikací pomocí vývojového prostředí Ionic. Tato volba se během implementace ukázala jako správná. Všechny potřebné funkce a služby, které bylo nutné během implementace aplikace využít, byly v Ionicu dostupné. Také všechny pluginy pro přístup k API telefonu využitě v aplikaci byly dostupné jako pluginy Ionicu. Vývojové prostředí Ionic je velmi dobrou volbou pro vývoj jednoduchých multiplatformních aplikací podobného typu, jako je vytvořená aplikace BRUNI.

Aplikace BRUNI je nyní dostupná v testovací verzi pro platformu Android přes aplikační obchod Google Play a pro platformu iOS přes testovací prostředí Ionic View. Aplikace zobrazuje základní informace o budovách Masarykovy Univerzity. Budovy jsou přehledně rozděleny do skupin podle organizačních jednotek univerzity. Aplikace dále umožňuje zobrazení budov na mapě včetně jejich skutečných půdorysů a spuštění navigace k vybrané budově pomocí aplikace Google Maps. Aplikace uživateli také nabízí možnost přidání budov mezi oblíbené. Oblíbené budovy jsou zobrazené na úvodní obrazovce aplikace a uživateli je tedy výrazně zjednodušen přístup k informacím vybraných budov.

V blízké budoucnosti se počítá s rozšířením aplikace o další funkce. Zejména je to doplnění tlačítka „Navigovat“ na obrazovku „Detail budovy“, které umožní spustit navigaci i v jiné navigační aplikaci, než Google Maps. Dále je to přidání fulltextového vyhledávání v datech o budovách a vytvoření nové obrazovky „Nastavení“ umožňující uživateli další nastavení aplikace.

8 SEZNAM POUŽITÉ LITERATURY

- [1] VÁCLAVÍK, Jan. *Jak vyvíjet mobilní aplikace* [online]. 2015 [cit. 2017-05-12]. Dostupné z: <http://janvaclavik.cz/jak-vyvijet-mobilni-aplikace/>.
- [2] HÁŠA, Filip. *Hybridní mobilní aplikace – kdy ano a kdy ne?* [online]. 2016 [cit. 2017-05-12]. Dostupné z: <https://pixelfield.cz/blog/hybridni-mobilni-aplikace-kdy-ano-a-kdy-ne/>.
- [3] MICHL, Petr. *Je lepší nativní aplikace nebo mobilní web?* [online]. 2012 [cit. 2017-05-12]. Dostupné z: http://www.m-journal.cz/cs/internet/je-lepsi-nativni-aplikace-nebo-mobilni-web-__s281x9241.html.
- [4] *Aplikace na ArcGIS Online* [online]. ARCDATA PRAHA, 2017 [cit. 2017-05-13]. Dostupné z: <https://www.arcdata.cz/produkty/arcgis/arcgis-online/aplikace>.
- [5] *Android Studio* [online]. Android.com, 2017 [cit. 2017-05-13]. Dostupné z: <https://developer.android.com/studio/index.html>.
- [6] SEMECKÝ, Vojtěch. *Android Studio – nové vývojové prostředí. Zdroják, o tvorbě webových stránek a aplikací* [online]. Devel.cz Lab, 2013 [cit. 2017-05-13]. Dostupné z: <https://www.zdrojak.cz/clanky/android-studio-nove-vyvojove-prostredi/>.
- [7] *Google představil Android Studio, IDE pro tvorbu aplikací* [online]. BusinessIT, 2013 [cit. 2017-05-13]. Dostupné z: <http://www.businessit.cz/cz/google-predstavil-android-studio-ide-pro-tvorbu-aplikaci.php>.
- [8] *Xcode* [online]. Apple, 2017 [cit. 2017-05-13]. Dostupné z: <https://developer.apple.com/xcode/ide/>.
- [9] *Integrované vývojové prostředí (IDE) sady Visual Studio* [online]. Microsoft, 2017 [cit. 2017-05-15]. Dostupné z: <https://www.visualstudio.com/cs/vs/>.
- [10] *Xamarin Platform* [online]. Xamarin, 2017 [cit. 2017-05-16]. Dostupné z: <https://www.xamarin.com/platform>.
- [11] *Building Cross Platform Applications* [online]. Xamarin, 2017 [cit. 2017-05-16]. Dostupné z: https://developer.xamarin.com/guides/cross-platform/application_fundamentals/building_cross_platform_applications/.
- [12] *Getting Started* [online]. Xamarin, 2017 [cit. 2017-05-16]. Dostupné z: https://developer.xamarin.com/guides/cross-platform/getting_started/.
- [13] *Apache Cordova Documentation* [online]. The Apache Software Foundation, 2015 [cit. 2017-05-16]. Dostupné z: <https://cordova.apache.org/docs/en/latest/>.
- [14] *Apache Cordova: Platform Support* [online]. The Apache Software Foundation, 2015 [cit. 2017-05-16]. Dostupné z: <https://cordova.apache.org/docs/en/latest/guide/support/index.html>.

- [15] CORRAL, Luis, Alberto SILLITTI a Giancarlo SUCCI. Mobile Multiplatform Development: An Experiment for Performance Analysis. *Procedia Computer Science* [online]. 2012(vol. 10), 736-743 [cit. 2017-05-18]. DOI: 10.1016/j.procs.2012.06.094. ISSN 18770509. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S1877050912004516>.
- [16] *PhoneGap Documentation* [online]. Adobe Systems, 2015 [cit. 2017-05-18]. Dostupné z: <http://docs.phonegap.com/>.
- [17] *Apache Cordova or Adobe PhoneGap? Know the Real Difference* [online]. Konstant Infosolutions, 2016 [cit. 2017-05-18]. Dostupné z: <http://www.businessofapps.com/apache-cordova-adobe-phonegap-know-real-differences/>.
- [18] *Ionic Documentation* [online]. Drifty Co, 2017 [cit. 2017-05-19]. Dostupné z: <https://ionicframework.com/docs/>.
- [19] VÁCLAVÍK, Jan. Vytvíjíme hybridní aplikace v Ionicu: Úvod a instalace. *Zdroják, o tvorbě webových stránek a aplikací* [online]. Devel.cz Lab, 2015 [cit. 2017-05-19]. Dostupné z: <https://www.zdrojak.cz/clanky/vyvijime-hybridni-aplikace-v-ionicu/>.
- [20] BANGA, Heena. Ionic + AngularJS: Transition from Native to Hybrid Mobile App Development. *Konstant Infosolutions* [online]. Konstant Infosolutions, 2016 [cit. 2017-05-19]. Dostupné z: <https://www.konstantinfo.com/blog/ionic-angularjs-transition-from-native-to-hybrid-mobile-app-development/>.
- [21] LYNCH, Max. *Announcing Ionic 2.0.0 Final* [online]. 2017 [cit. 2017-05-19]. Dostupné z: <http://blog.ionic.io/announcing-ionic-2-0-0-final/>.
- [22] *React Native: Docs* [online]. Facebook, 2017 [cit. 2017-05-19]. Dostupné z: <https://facebook.github.io/react-native/releases/next/docs/getting-started.html>.
- [23] *React Native - Overview* [online]. Tutorialspoint.com, 2017 [cit. 2017-05-19]. Dostupné z: https://www.tutorialspoint.com/react_native/react_native_overview.htm.
- [24] *Mobile App Development & MBaaS / Axway Appcelerator* [online]. Axway, 2017 [cit. 2017-05-19]. Dostupné z: <https://www.appcelerator.com/>.
- [25] *Ext JS - JavaScript framework for web apps* [online]. Sencha, 2017 [cit. 2017-05-19]. Dostupné z: <https://www.sencha.com/products/extjs/#overview>.
- [26] *AppStudio for ArcGIS* [online]. Environmental Systems Research Institute, 2017 [cit. 2017-05-19]. Dostupné z: <https://appstudio.arcgis.com/>.
- [27] PSOTA, Karel. AppStudio for ArcGIS: vytváříme nativní aplikace. *Arcrevue: časopis pro uživatele softwaru ESRI a ENVI*. Praha: Arcdata, 2015(3), 28-29. ISSN 1211-2135.

9 SEZNAM ZKRATEK, SYMBOLŮ, OBRÁZKŮ A TABULEK

9.1 Seznam zkratek

- **API** (Application Programming Interface) – označuje rozhraní pro programování aplikací,
- **CSS** (Cascading Style Sheets) – kaskádové styly – jazyk pro popis způsobu zobrazení elementů na stránkách napsaných v HTML,
- **GIS** (Geografický informační systém) – systém umožňující ukládat, spravovat a analyzovat prostorová data,
- **HTML** (HyperText Markup Language) – značkovací jazyk pro tvorbu webových stránek,
- **IDE** (Integrated Development Environment) – vývojové prostředí,
- **JS** (JavaScript) – skriptovací programovací jazyk,
- **JSON** (JavaScript Object Notation) - způsob zápisu dat nezávislý na počítačové platformě,
- **OS** – operační systém,
- **SDK** (Software development kit) – sada vývojových nástrojů pro vytváření aplikací,
- **SQL** (Structured Query Language) – standardizovaný strukturovaný dotazovací jazyk, který je používán pro práci s daty v relačních databázích,
- **UI** (User Interface; GUI = Graphical UI) – (grafické) uživatelské rozhraní,
- **UI prvek** – interaktivní grafický ovládací prvek uživatelského rozhraní.

9.2 Seznam obrázků

- Obrázek 1 – Ukázka vývojového prostředí Android Studio 2.3 – Design mód (str. 23)
- Obrázek 2 – Ukázka vývojového prostředí Xcode – Interface Builder (str. 25)
- Obrázek 3 – Ukázka vývojového prostředí Visual Studio 2017 (str. 26)
- Obrázek 4 – Sdílení kódu v Xamarinu (str. 27)
- Obrázek 5 – Schéma architektury Cordova aplikace (str. 28)
- Obrázek 6 – Ionic Creator (str. 33)
- Obrázek 7 – Ukázka prostředí AppStudio for ArcGIS Web Edition (str. 36)
- Obrázek 8 – Aplikace Budovy MU (str. 39)
- Obrázek 9 – Aplikace MUNI Buildings (str. 41)
- Obrázek 10 – Aplikace Easy Brno (str. 42)
- Obrázek 11 – Logika přechodů mezi obrazovkami (str. 44)

- Obrázek 12 – Implementovaná aplikace BRUNI (str. 55)

9.3 Seznam tabulek

- Tabulka 1 – Vhodné a nevhodné typy aplikací pro konkrétní přístupy k vývoji (str. 20)
- Tabulka 2 – Apache Cordova – podporované platformy (str. 28)
- Tabulka 3 – Apache Cordova – Core plugins (str. 30)

10 SEZNAM PŘÍLOH

- CD:
 - APK – aplikační balíček aplikace BRUNI pro platformu Android
 - Bruni – zdrojový Ionic projekt aplikace BRUNI
 - DB – vzorová SQL databáze
 - DP – text diplomové práce
 - Server – zdrojové kódy serverových služeb
- Příloha 1 – Ukázka zdrojového kódu pro obrazovku „Skupiny“ (str. 69)
- Příloha 2 – Ukázka zdrojového kódu serverové služby „metadata“ (str. 71)
- Příloha 3 - Výsledná aplikace BRUNI (str. 73)

PŘÍLOHY

Příloha 1 – Ukázka zdrojového kódu pro obrazovku „Skupiny“ Kontroler (JavaScript):

```
1  // Kontroler - Skupiny
2  app.controller('skupinyCtrl',
3  function ($scope, $state, $translate, LokiService) {
4
5      // Inicializace stranky
6      function init_page() {
7          $scope.lang = $translate.proposedLanguage();
8
9          $scope.loc_db = [];
10         $scope.loc_db["config"] = LokiService.config;
11         $scope.loc_db["buildings"] = LokiService.buildings;
12         $scope.loc_db["groups"] = LokiService.groups;
13     }
14
15     // Pri kazdem otevreni stranky
16     $scope.$on('$ionicView.enter', function(scopes, states) {
17         console.log("\nSCOPE ON - skupiny");
18         init_page();
19     })
20
21     // Prepnuti stranky
22     $scope.swipePage = function (state_name) {
23         $state.go(state_name);
24     }
25 });
26
```

Template (HTML):

```
1  <ion-view title="{{'tabs.skupiny' | translate}}">
2      <ion-nav-bar class="bar-royal"></ion-nav-bar>
3
4      <ion-content padding="false" class="has-header">
5
6          <div class="list">
7              <a class="item my-group-item" ng-repeat="group in loc_db.groups.data |
8                  orderBy : 'nazev_cz'" ng-show="lang=='cz' || lang=='cs'" ui-sref="
9                  app.tabs.budovy({'tag': group.tag})">
10
11                  <table>
12                      <tr>
13                          <td>
14                              
15                              </img>
16                          </td><td class="my-group-name">
17                              {{group.nazev_cz}}
18                          </td>
19                      </tr>
20                  </table>
21              </a>
22              ...
23          </div>
24
25      </ion-content>
26  </ion-view>
27
```


Příloha 2 – Ukázka zdrojového kódu serverové služby „metadata“

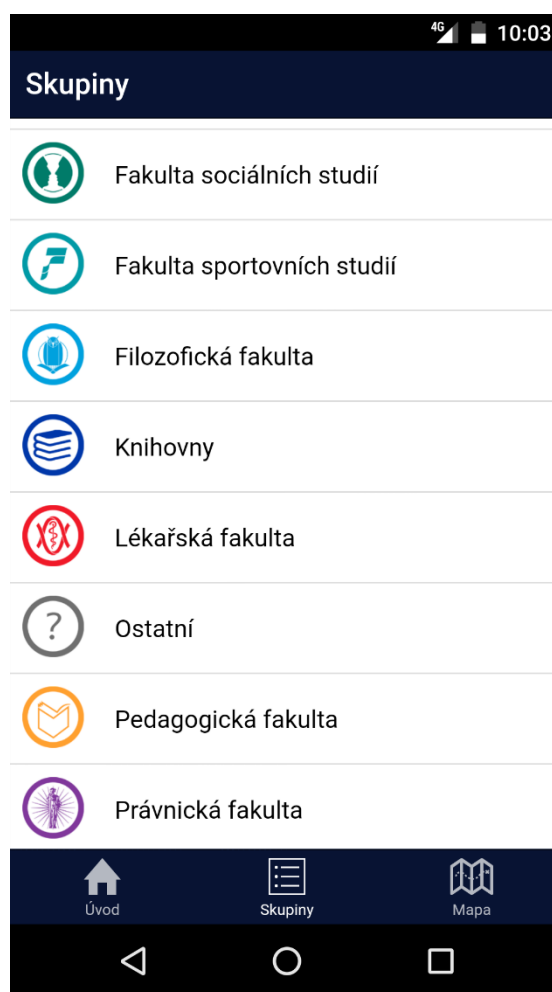
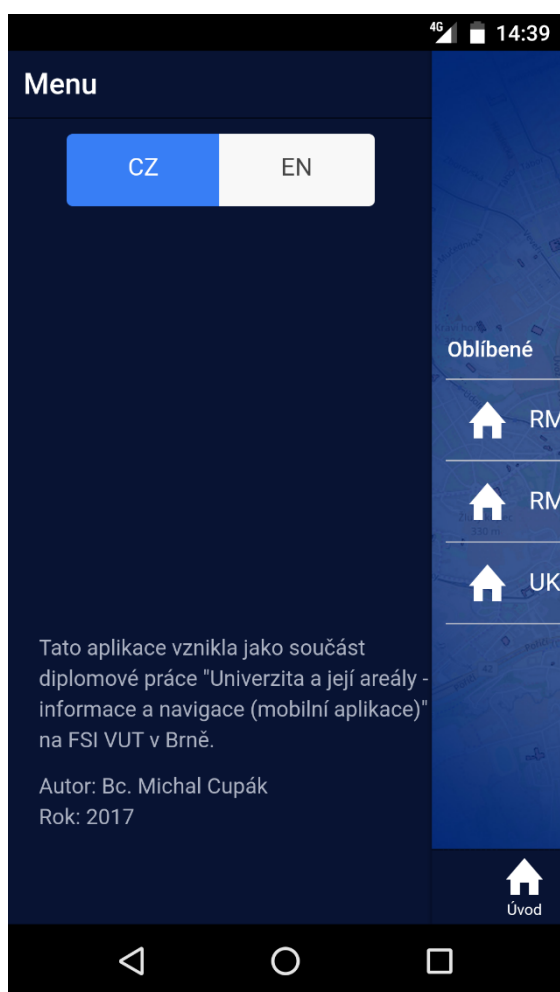
```
1 <?php
2
3 header('Access-Control-Allow-Origin: *');
4 header('Content-Type: application/json; charset=utf-8');
5 header("Cache-Control: no-store, no-cache, must-revalidate, max-age=0");
6 header("Cache-Control: post-check=0, pre-check=0", false);
7 header("Pragma: no-cache");
8
9 $output = array();
10 $output["state"] = 0;
11
12 ob_start();
13
14 $servername = "wm133.wedos.net";
15 $username = "w152088_bruni";
16 $password = "vxQN7Tbf";
17 $dbname = "d152088_bruni";
18
19 // Create connection
20 $conn = new mysqli($servername, $username, $password, $dbname);
21
22 // Check connection
23 if (!$conn->connect_error) {
24     //die("Connection failed: " . $conn->connect_error);
25
26     $conn->set_charset("utf8");
27
28     $sql = "SELECT * FROM metadata ORDER BY id DESC LIMIT 1;";
29     $result = $conn->query($sql);
30
31     if ($result->num_rows > 0) {
32         // output data
33         $output["state"] = 1;
34         $output["data"] = $result->fetch_assoc();
35     }
36     $conn->close();
37 }
38
39 ob_end_clean();
40
41 $json = json_encode($output);
42 print($json);
43
44 ?>
45
```


Příloha 3 - Výsledná aplikace BRUNI

Obrazovka „Úvod“:



Boční menu a obrazovka „Skupiny“:



Obrazovka „Budovy“ a obrazovka „Detail budovy“:



Obrazovka „Mapa“:

